



Universidad
Carlos III de Madrid

*“Técnicas de computación evolutiva
aplicadas a la clasificación a partir de
monitores de actividad física”*

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autora:

María del Carmen Del
Barrio Cerro

Tutores:

Yago Sáez Achaerandio
Alejandro Baldominos

Marzo 2016

Índice de Contenido

Lista de tablas	6
Lista de figuras	12
Resumen	13
Abstract	15
Agradecimientos	17
1. Introducción	18
1.1 Motivación	18
1.2 Objetivos	20
1.3 Estructura del documento	21
2. Planteamiento del problema	23
2.1 Estado del arte	23
2.1.1 Monitorización	24
2.1.2 Reconocimiento de actividades	26
2.1.3 Trabajos relacionados	40
2.2 Técnicas	43
2.2.1 Aprendizaje automático	43
2.2.2 Computación evolutiva	49

2.3 Análisis.....	65
2.3.1 Requisitos de usuario	65
3. Diseño e implementación.....	71
3.1 Funcionamiento general de los módulos	72
3.2 Diagramas de flujo	73
3.3 Diagramas de clases.....	80
3.4 Implementación.....	84
3.4.1 Optimización de clustering: enjambre de partículas	84
3.4.2 Aprendizaje de clasificación: algoritmo evolutivo	88
3.4.3 Evaluación del reconocimiento de actividades	93
4. Experimentación	96
4.1 Metodología	96
4.1.1 Adquisición de datos.....	97
4.1.2 Preprocesamiento de datos	98
4.1.3 Segmentación de los datos.....	100
4.1.4 Selección de atributos.....	101
4.1.5 Entrenamiento y clasificación.....	102
4.2 Configuración de parámetros	103
4.2.1 Enjambre de partículas	103

4.2.2 Algoritmo evolutivo	106
4.3 Experimentos	108
4.3.1 Experimentos independientes del sujeto.....	109
4.3.2 Experimentos dependientes del sujeto	127
4.3.3 Experimentos parcialmente dependientes del sujeto	144
4.4 Resumen y análisis	148
4.4.1 Experimentos independientes del sujeto.....	148
4.4.2 Experimentos dependientes del sujeto	150
4.4.3 Experimentos parcialmente dependientes del sujeto	150
5. Planificación	152
5.1 Tareas.....	152
5.2 Planificación inicial	153
5.3 Planificación final	156
6. Marco regulador.....	159
7. Presupuesto	161
7.1 Costes de personal	161
7.2 Costes de software	162
7.3 Costes de hardware	163
7.4 Otros costes	163

7.5 Coste total	164
8. Conclusiones	165
8.1 Cumplimiento de objetivos	165
8.2 Líneas futuras de trabajo	166
ANEXO I. Competencias en inglés	168
Introduction	168
Objectives	170
Design of the solution.....	171
General procedure of the modules	172
Experiments and results.....	174
Conclusions	177
ANEXO II. Árboles	180
Referencias	181

Lista de tablas

Tabla 1: Trabajos relacionados.....	42
Tabla 2: Genérica de requisitos	65
Tabla 3: Requisitos de capacidad	66
Tabla 4: Tiempo de realización de actividades	98
Tabla 5: Enjambre de partículas	103
Tabla 6: Combinación de valores.....	104
Tabla 7: Algoritmo evolutivo.....	106
Tabla 8: Matriz de confusión	108
Tabla 9: Matriz de confusión	110
Tabla 10: tabla de clases con precisión y cobertura.....	110
Tabla 11: Matriz de confusión	111
Tabla 12: tabla de clases con precisión y cobertura.....	111
Tabla 13: Matriz de confusión	112
Tabla 14: tabla de clases con precisión y cobertura.....	112
Tabla 15: Matriz de confusión	113
Tabla 16: tabla de clases con precisión y cobertura.....	113
Tabla 17: Matriz de confusión	114

Tabla 18: tabla de clases con precisión y cobertura.....	114
Tabla 19: Matriz de confusión	115
Tabla 20: tabla de clases con precisión y cobertura.....	115
Tabla 21: Matriz de confusión	116
Tabla 22: tabla de clases con precisión y cobertura.....	116
Tabla 23: Matriz de confusión	117
Tabla 24: tabla de clases con precisión y cobertura.....	117
Tabla 25: tasa de acierto de los sujetos y media por procedimiento básico	118
Tabla 26: tasa de acierto de los sujetos y media por procedimiento con probabilidades de pertenencia	118
Tabla 27: Matriz de confusión	118
Tabla 28: tabla de clases con precisión y cobertura.....	119
Tabla 29: Matriz de confusión	119
Tabla 30: tabla de clases con precisión y cobertura.....	120
Tabla 31: Matriz de confusión	120
Tabla 32: tabla de clases con precisión y cobertura.....	121
Tabla 33: Matriz de confusión	121
Tabla 34: tabla de clases con precisión y cobertura.....	122
Tabla 35: Matriz de confusión	122

Tabla 36: tabla de clases con precisión y cobertura.....	123
Tabla 37: Matriz de confusión	123
Tabla 38: tabla de clases con precisión y cobertura.....	124
Tabla 39: Matriz de confusión	124
Tabla 40: tabla de clases con precisión y cobertura.....	125
Tabla 41: Matriz de confusión	125
Tabla 42: tabla de clases con precisión y cobertura.....	126
Tabla 43: tasa de acierto de los sujetos y media por procedimiento básico	126
Tabla 44: tasa de acierto de los sujetos y media por procedimiento con probabilidades de pertenencia	126
Tabla 45: tasa de acierto de las ejecuciones y media	127
Tabla 46: precisión y cobertura del mejor <i>fold</i> por clases	127
Tabla 47: matriz de confusión del mejor <i>fold</i>	128
Tabla 48: tasa de acierto de las ejecuciones y media	128
Tabla 49: precisión y cobertura del mejor <i>fold</i> por clases	128
Tabla 50: matriz de confusión del mejor <i>fold</i>	129
Tabla 51: tasa de acierto de las ejecuciones y media	129
Tabla 52: precisión y cobertura del mejor <i>fold</i> por clases	129
Tabla 53: matriz de confusión del mejor <i>fold</i>	130

Tabla 54: tasa de acierto de las ejecuciones y media	130
Tabla 55: precisión y cobertura del mejor <i>fold</i> por clases	130
Tabla 56: matriz de confusión del mejor <i>fold</i>	131
Tabla 57: tasa de acierto de las ejecuciones y media	131
Tabla 58: precisión y cobertura del mejor <i>fold</i> por clases	131
Tabla 59: matriz de confusión del mejor <i>fold</i>	132
Tabla 60: tasa de acierto de las ejecuciones y media	132
Tabla 61: precisión y cobertura del mejor <i>fold</i> por clases	132
Tabla 62: matriz de confusión del mejor <i>fold</i>	133
Tabla 63: tasa de acierto de las ejecuciones y media	133
Tabla 64: precisión y cobertura del mejor <i>fold</i> por clases	133
Tabla 65: matriz de confusión del mejor <i>fold</i>	134
Tabla 66: tasa de acierto de las ejecuciones y media	134
Tabla 67: precisión y cobertura del mejor <i>fold</i> por clases	134
Tabla 68: matriz de confusión del mejor <i>fold</i>	135
Tabla 69: tasa de acierto de las ejecuciones y media	135
Tabla 70: precisión y cobertura del mejor <i>fold</i> por clases	135
Tabla 71: matriz de confusión del mejor <i>fold</i>	136
Tabla 72: tasa de acierto de las ejecuciones y media	136

Tabla 73: precisión y cobertura del mejor <i>fold</i> por clases	136
Tabla 74: matriz de confusión del mejor <i>fold</i>	137
Tabla 75: tasa de acierto de las ejecuciones y media	137
Tabla 76: precisión y cobertura del mejor <i>fold</i> por clases	137
Tabla 77: matriz de confusión del mejor <i>fold</i>	138
Tabla 78: tasa de acierto de las ejecuciones y media	138
Tabla 79: precisión y cobertura del mejor <i>fold</i> por clases	138
Tabla 80: matriz de confusión del mejor <i>fold</i>	139
Tabla 81: tasa de acierto de las ejecuciones y media	139
Tabla 82: precisión y cobertura del mejor <i>fold</i> por clases	139
Tabla 83: matriz de confusión del mejor <i>fold</i>	140
Tabla 84: tasa de acierto de las ejecuciones y media	140
Tabla 85: precisión y cobertura del mejor <i>fold</i> por clases	140
Tabla 86: matriz de confusión del mejor <i>fold</i>	141
Tabla 87: tasa de acierto de las ejecuciones y media	141
Tabla 88: precisión y cobertura del mejor <i>fold</i> por clases	141
Tabla 89: matriz de confusión del mejor <i>fold</i>	142
Tabla 90: tasa de acierto de las ejecuciones y media	142
Tabla 91: precisión y cobertura del mejor <i>fold</i> por clases	142

Tabla 92: matriz de confusión del mejor <i>fold</i>	143
Tabla 93: tasa de acierto media de cada sujeto y media total por procedimiento básico.....	143
Tabla 94: tasa de acierto media de cada sujeto y media total por procedimiento con probabilidades de pertenencia	143
Tabla 95: tasa de acierto de las ejecuciones y media	144
Tabla 96: precisión y cobertura del mejor <i>fold</i> por clases	144
Tabla 97: matriz de confusión del mejor <i>fold</i>	145
Tabla 98: tasa de acierto de las ejecuciones y media	145
Tabla 99: precisión y cobertura del mejor <i>fold</i> por clases	145
Tabla 100: matriz de confusión del mejor <i>fold</i>	146
Tabla 101: tasa de acierto de las ejecuciones y media	146
Tabla 102: precisión y cobertura del mejor <i>fold</i> por clases.....	146
Tabla 103: matriz de confusión del mejor <i>fold</i>	147
Tabla 104: tasa de acierto de las ejecuciones y media	147
Tabla 105: precisión y cobertura del mejor <i>fold</i> por clases.....	147
Tabla 106: matriz de confusión del mejor <i>fold</i>	148
Tabla 107: Planificación inicial.....	154
Tabla 108: Planificación final	157
Tabla 109: Costes de personal	161

Tabla 110: Costes de software	162
Tabla 111: Costes de hardware	163
Tabla 112: Coste total.....	164

Lista de figuras

Ilustración 1: Cadena de Reconocimiento de Actividades.....	30
Ilustración 2: Mecanismo del sistema	71
Ilustración 3: Diagrama de flujo Módulo 1: PSO.....	73
Ilustración 4: Diagrama de flujo Módulo 2: AE	76
Ilustración 5: Diagrama de flujo Módulo 3: PEv.....	79
Ilustración 6: Diagrama de clases Módulo 1: PSO	81
Ilustración 7: Diagrama de clases Módulo 2: AE.....	82
Ilustración 8: Diagrama de clases Módulo 3: PEv	83
Ilustración 9: Peso inercia.....	105
Ilustración 10: Peso media.....	106
Ilustración 11: Planificación inicial.....	155
Ilustración 12: Planificación final	158

Resumen

Actualmente, varios factores están haciendo que el campo de reconocimiento de actividades humanas cobre una mayor importancia, como por ejemplo, la proliferación de dispositivos “wearables” que permiten monitorizar la actividad física o la tendencia de la población mundial hacia un estilo de vida cada vez más sedentario. Este estilo de vida sedentario propio de la sociedad actual se traduce en insuficiente actividad física y se considera uno de los mayores factores de riesgo para la salud, estando entre los primeros puestos de factores de riesgo de mortalidad a nivel mundial, según la OMS [11].

De esta manera, dentro del ámbito de la salud y el bienestar, gracias al avance en la miniaturización de sensores, que incluso permite su uso incorporados a la ropa de las personas, el reconocimiento automático de actividades se presenta como una solución a problemas de diversa índole, como por ejemplo, prevención de enfermedades, envejecimiento activo, monitorización remota de enfermos, además de un amplio espectro de aplicaciones en el ámbito deportivo.

Es por esto que se convierten en dispositivos de monitorización sumamente útiles en otras áreas de investigación, introduciendo el reconocimiento de actividades humanas en la computación ubicua, el entretenimiento, el registro de actividades diarias personales o el seguimiento del rendimiento deportivo o profesional.

Con la principal motivación de explorar nuevos frentes de investigación del reconocimiento de actividades, con un enfoque distinto a los planteados hasta ahora, en este trabajo se propone un sistema de reconocimiento automático de actividades que integra un algoritmo evolutivo, para la tarea de clasificación de actividades, y un enjambre de partículas, para la realización de un *clustering* que mejore el aprendizaje automático.

El sistema ha sido evaluado mediante validación cruzada del tipo *leave-one-subject-out*, para comprobar su rendimiento en situaciones de reconocimiento independiente del sujeto, obteniendo un **52,37%** de acierto.

También, se ha evaluado el sistema con validación cruzada estándar de *10-folds* en cada sujeto, para analizar la capacidad del sistema en casos de clasificación dependiente del sujeto, alcanzando un **98,07%** de acierto.

Un resultado significativamente más positivo que el primero, que muestra que el sistema puede tender a la personalización del reconocimiento de actividades.

Además, se ha llevado a cabo la evaluación del sistema con validación cruzada estándar de *10-folds* en el conjunto de todos los sujetos, con un **70,2267%** de acierto, abundándose en la conclusión expuesta más arriba, de que el sistema presenta un mejor funcionamiento en situaciones de personalización del reconocimiento de actividades.

Palabras clave: reconocimiento de actividades; actividad física; monitorización; computación evolutiva; algoritmo evolutivo; optimización con enjambre de partículas; coevolución; aprendizaje automático; clasificación; clustering.

Abstract

In the current time, various factors are making the field of activity recognition become more important, such as the proliferation of wearable devices that allow to monitor physical activity or global population's tendency towards a more sedentary lifestyle. This sedentary lifestyle is turning into insufficient physical activity and is considered one of the factors with a highest risk for health, being among the leading risk factors of mortality, regarding the WHO [11].

This way, within health and wellness field, thanks to the advance in sensor miniaturization, which even allows sensor usage incorporated to people clothes, activity automatic recognition is presented as a solution to very diverse problems, such as diseases prevention, active aging, patient remote monitoring, as well as a wide range of applications in sports.

For that reason, wearable sensors happen to be extremely useful monitoring devices in other research areas, introducing human activity recognition to ubiquitous computing, entertainment industry, daily life activities logging and sportive and professional performance monitoring, among others.

With the main motivation of exploring new research horizons, through a different approach to the previous works, in this project, an activity automatic recognition system that integrates an evolutionary algorithm, for the activity classification task, and a particle swarm, for a clustering that improves the automatic learning, is proposed.

The system has been evaluated with leave-one-subject-out (LOSO) cross validation, in order to assess its performance in situations where the recognition is subject independent, obtaining an accuracy rate of **52,37%**.

Also, the system has been evaluated with 10-fold standard cross validation within each subject, to analyze the system's capacity in subject dependent classification cases, reaching an accuracy rate of **98,07%**. A significantly more positive result than the first one, that shows the system might tend to personalization of activity recognition.

In addition, the system evaluation has been carried with 10-fold standard cross validation within the whole set of all the subjects, getting an

accuracy rate of **70,2267%**, which supports the conclusion presented above that the system works better in situations of personalization of the activity recognition.

Key words: activity recognition; physical activity; monitoring; evolutionary computing; evolutionary algorithm; particle swarm optimization; coevolution; machine learning; classification; clustering.

Agradecimientos

En primer lugar, me gustaría dar las gracias a mi familia, por apoyarme a la par que hacerme reaccionar cuando era conveniente, y, en especial, a mi madre, por hacer las veces de psicóloga, animadora, ayudante y mil cosas más.

También, quiero agradecer a mis amigos por hacerme desconectar y recargar mi energía cuando más falta me hacía.

Por supuesto, me gustaría dar las gracias a mis tutores, Yago y Alejandro, por depositar toda su confianza en mí, y, en particular a Yago, por darme la oportunidad de realizar el trabajo de fin de grado sobre un tema que realmente ha despertado mi interés.

Por último, pero no menos importante, debo agradecer infinitamente a mis compañeros, y amigos, de carrera, que han hecho del camino hasta aquí lo más insospechado. Sin ellos, me habría perdido momentos únicos, sin ellos, nunca una sesión de estudio pre-examen habría acabado tornándose en una clase magistral propia de ingeniería de cordones, con el fin de rescatar un desafortunado bolígrafo atrapado en el interior de la pata de una mesa.

1. Introducción

1.1 Motivación

Hoy en día, el reconocimiento automático de actividades tiene múltiples campos de aplicación, concretamente en el ámbito de la salud y el bienestar, esta disciplina se emplea con una extensa variedad de fines, pero también en otras muchas áreas, como la computación ubicua, el entretenimiento, el registro diario personal de actividades o el seguimiento del rendimiento en contextos deportivos o profesionales.

La población humana tiende hacia un estilo de vida cada vez más sedentario, cobrando cada vez mayor importancia la investigación de los vínculos entre la inactividad física y sus consecuencias, desde enfermedades no transmisibles (ENT), como enfermedades cardiovasculares [141], hipertensión [142], diabetes [143], hasta trastornos de carácter psicológico como la depresión [144].

La Organización Mundial de la Salud (OMS), sostiene unas cifras y hechos que reflejan la seriedad de esta propensión de la población, la reacción generada al respecto y su relación con algunas afecciones de la salud:

- La inactividad física ocupa el cuarto lugar entre los principales factores de riesgo de mortalidad a nivel mundial.
- Aproximadamente 3,2 millones de personas mueren cada año debido a la inactividad física.
- La inactividad física es uno de los principales factores de riesgo de padecer enfermedades no transmisibles (ENT), como las enfermedades cardiovasculares, el cáncer y la diabetes.
- La actividad física tiene importantes beneficios para la salud y contribuye a prevenir las ENT.
- A nivel mundial, uno de cada tres adultos no tiene un nivel suficiente de actividad física.

- El 56% de los Estados Miembros de la OMS ha puesto en marcha políticas para reducir la inactividad física.
- Los Estados Miembros de la OMS han acordado reducir la inactividad física en un 10% para 2025.

Ante esta tendencia, se presentan los sistemas de reconocimiento de actividades como respuesta a la necesidad de métodos automáticos para cuantificar rutinas y patrones de actividades humanas, aplicándose en la investigación para iniciativas que promuevan la actividad física [57, 136], la mejora del diagnóstico y tratamiento de trastornos neurológicos, degenerativos y respiratorios, como el Parkinson [137, 138], la esclerosis múltiple [139] o enfermedades pulmonares crónicas [140].

Además, los recientes avances en la miniaturización de sensores, permiten recopilar datos sobre muchos aspectos del movimiento humano de manera cada vez más sencilla y autónoma, abriendo nuevos horizontes, incluyendo el uso de sensores en las personas (*wearables*), o para la utilización del reconocimiento de actividades en sistemas de computación ubicua, entre otros.

Sin embargo, todos esos datos repletos de información potencialmente útil, carecen de significado práctico si no son interpretados por algoritmos capaces de extraer el conocimiento, en caso que nos ocupa, referente a las actividades humanas involucradas. El desarrollo de sistemas de reconocimiento de actividades cuenta con múltiples desafíos a superar y resolver, siendo un campo de trabajo atractivo, aparte de las ya mencionadas diversas aplicaciones.

Por último, a pesar de que el estado del arte de los mecanismos de reconocimiento de actividades cuenta con numerosas propuestas de naturaleza muy diversa, es un campo relativamente nuevo, y los planteamientos que incluyen técnicas de computación evolutiva, aunque son prometedores [145][146], no son abundantes. No obstante, el problema que se plantea en este trabajo, aunque con diversas cuestiones particulares de este ámbito de aplicación (ruido, big-data, etc.), se ubica dentro de la clasificación mediante técnicas de aprendizaje automático, campo que si está ampliamente explorado [243].

1.2 Objetivos

Siguiendo la línea de lo expuesto hasta aquí, los objetivos que presenta este trabajo están más ligados a la investigación, con el fin de abrir nuevos frentes de estudio mediante un enfoque alternativo para el reconocimiento de actividades, que al desarrollo de una aplicación o sistema de uso práctico o potencialmente comercializable de cara al público y a los usuarios de este sector. Por consiguiente, las metas propuestas para este trabajo son las siguientes:

- La implementación y el empleo de técnicas de computación evolutiva. De este punto se derivan otros dos más específicos:
 - El diseño y la utilización de un algoritmo evolutivo para la clasificación de actividades.
 - El desarrollo de un enjambre de partículas para la optimización del clustering previo de los datos.
- La integración de la coevolución, a ambas partes del sistema, con el fin de agilizar la convergencia y obtener el máximo partido a los datos disponibles en la fase de aprendizaje.
- La utilización de un conjunto de datos público, ya propuesto en otros trabajos que permita comparar los resultados obtenidos y que contenga datos recogidos mediante sensores *wearable*.
- Evaluación del sistema implementado en base a tres enfoques distintos, según el nivel de especialización o generalización del clasificador en relación a los usuarios.
- La obtención de resultados que, aún pudiendo ser mejorables por investigadores expertos, sean comparables al estado del arte, resultando satisfactorios y fomentado así continuar por esta línea de investigación.
- La realización de un estudio del estado del arte que abarque tanto el reconocimiento de actividades, como la computación evolutiva y biológica y el aprendizaje automático, dentro de éste, la clasificación y la agrupación o clustering, concretamente.

Además, a un nivel académico más personal, este trabajo supone la persecución de ciertos objetivos adicionales:

- La adquisición de nuevos conocimientos en las áreas de reconocimiento de actividades, computación evolutiva y aprendizaje automático.

- La ampliación de la experiencia en la utilización de técnicas de computación evolutiva, de clasificación y clustering, así como de la metodología de reconocimiento de actividades humanas.
- El desarrollo de un proyecto en su totalidad, incluyendo la toma de decisiones pertinentes, el uso de las herramientas adecuadas y la aplicación práctica de los conocimientos teóricos, además de la documentación completa y rigurosa del mismo.

1.3 Estructura del documento

Este documento recoge todos los aspectos relevantes del trabajo llevado a cabo, y la estructura seguida en el mismo se explica a continuación.

Al inicio del documento, se encuentra el índice de contenidos, seguido por las listas de figuras y tablas, así como el resumen del trabajo, el abstract, o versión del resumen en inglés, y los agradecimientos.

El capítulo 1, en el que se encuentra el lector, contiene una introducción al trabajo, con la motivación y los objetivos que han llevado a su realización, y también el presente apartado.

Seguidamente, el capítulo 2 consiste en el planteamiento del problema, donde se incluye el estudio del estado del arte del reconocimiento de actividades y las técnicas de computación evolutiva y de aprendizaje automático, además del correspondiente análisis y los requisitos extraídos del mismo.

En el capítulo 3, se expone el diseño de los módulos a implementar y, a continuación, la funcionalidad de los mismos, una vez implementada.

Por su parte, el capítulo 4, describe la experimentación llevada a cabo para la evaluación del sistema de clasificación implementado, así como la metodología seguida, la configuración de parámetros y las pruebas y sus resultados en detalle.

Después, el capítulo 5 presenta la planificación del proyecto y las tareas correspondientes, el capítulo 6 explica diferentes aspectos del marco regulador, y el capítulo 7 recoge un presupuesto detallado referente a la realización del proyecto.

Finalmente, en el capítulo 8 se exponen las conclusiones derivadas de todas las fases efectuadas en el transcurso de este trabajo. A continuación del mismo, se pueden encontrar dos anexos, uno que contiene un resumen relativamente amplio redactado en inglés, destinado a demostrar las competencias en inglés, y otro que recoge las diferentes representaciones de un árbol de decisión, implicadas en el trabajo. Por último, se listan las referencias utilizadas e incluidas en el trabajo.

2. Planteamiento del problema

Para un mayor entendimiento del problema que atañe a este proyecto, se expone un estudio sobre los diferentes campos de investigación relacionados y, a partir de la misma, el análisis realizado para determinar los requisitos y encauzar la resolución a desarrollar.

En primer lugar, se explica el estado del arte con respecto a la monitorización y reconocimiento de actividades humanas, incluyendo los dispositivos involucrados en la recolección de datos.

En segundo lugar, se describen las distintas técnicas, tanto de aprendizaje automático como de computación evolutiva, implicadas en la solución implementada.

Por último, se concluye con la especificación de los requisitos tomados para el desarrollo del sistema.

2.1 Estado del arte

Durante la pasada década, la microelectrónica y los sistemas de computación han experimentado un desarrollo excepcional, capacitando de características sin precedentes, que combinan alta potencia, y tamaño y coste reducido, tanto a sensores como dispositivos móviles, haciendo accesible su uso diario. El avance de los sensores ubicuos va de la mano de la creciente actividad del área de investigación centrada en la extracción de conocimiento a partir de la información recogida por estos sensores, la llamada computación ubicua o pervasiva (*ubiquitous/pervasive computing*) [2].

Concretamente, dentro de este campo ha despertado un gran interés el reconocimiento de actividades humanas, sobre todo, su aplicación a servicios de carácter médico, militar y de seguridad. Disciplina que, a pesar de iniciarse a finales de los años noventa [3, 4], aún presenta muchos aspectos a desarrollar para aumentar el rendimiento de sus técnicas en situaciones lo más realistas posible.

Antes de ampliar el estado del arte del reconocimiento de actividades, se expone un breve estudio sobre la monitorización de las mismas, así como los dispositivos utilizados para ello y, particularmente, las unidades de medición inercial (*inertial measurement units, IMUs*) dentro de los sensores que puede portar una persona (*wearable, body-worn sensors*).

2.1.1 Monitorización

La monitorización de actividades humanas, incluso de carácter diario, es una práctica muy extendida hoy en día gracias al avance de las tecnologías de sensores, que permiten un uso prácticamente desapercibido de los mismos, como es el caso de los dispositivos *wearable* [13]. La monitorización se realiza mediante numerosas técnicas, y las enfocadas a la monitorización personal se engloban dentro del movimiento conocido como “yo cuantificado” (*quantified self*), que consiste en el conocimiento de uno mismo mediante su propio seguimiento. Para ello, se incorpora la tecnología, normalmente, sensores y computación *wearable*, en la vida diaria de una persona, recolectando información sobre datos cuantificables, estados y rendimiento. Los términos “yo cuantificado” fueron propuestos por Gary Wolf y Kevin Kelly en 2007 [5], pero este concepto recibe otros muchos nombres, como *lifelogging* [6], *self-tracking* [7], *auto-analytics*, *body-hacking*, *self-quantifying*, etc. Este tipo de monitorización se aplica a diversos aspectos de la vida con el objetivo de mejorarlos, destacando su aportación en el campo de la salud y el bienestar [8, 9], a partir del seguimiento de factores como la actividad física, el consumo de calorías, la calidad de sueño y la postura del cuerpo, entre otros. También son campos de aplicación del “yo cuantificado” el aumento de productividad tanto personal como profesional [10], incluyendo el rendimiento deportivo [12], y la gamificación de actividades corrientes. Algunos de los monitores de actividad personal que se utilizan en la actualidad son: Apple Watch, Fitbit Tracker, Jawbone UP, Nike+ FuelBand, Samsung Gear Fit o Misfit Wearables.

Dentro de los sensores que un individuo puede llevar, destacan por su uso mayoritario los sensores inerciales, sobre todo, los acelerómetros y giroscopios, pudiendo complementarse con magnetómetros [19], para medir la orientación del eje vertical con mayor precisión [20], o con GPS [21], para la localización.

Las unidades de medición inercial o IMUs proporcionan unos parámetros que, basándose en la hipótesis de la biomecánica de la descomposición en segmentos del cuerpo humano, son buenos recursos para el análisis de las cadenas cinéticas corporales, y su aplicación al

análisis de patologías, programas de rehabilitación y estudios deportivos [14, 15, 16, 17]. Además, tras los recientes avances tecnológicos y metodológicos en microsistemas electromecánicos (*micro-electromechanical systems, MEMS*), el precio, tamaño y portabilidad de las IMUs se ha visto reducido, facilitando y volviendo asequibles sus aplicaciones, así como abriendo nuevas líneas de investigación, especialmente esperanzadoras en el área de la biomecánica.

En la etapa inicial de estos dispositivos, donde fue determinante su desarrollo tecnológico, se produjeron dos progresos fundamentales: la disminución gradual del tamaño que permitió el aumento del número de sensores integrados, y la mejora del posprocesamiento de las mediciones de velocidad angular y aceleración lineal, corrigiéndose los errores, como los derivados del ruido, mediante sofisticados filtros, técnicas de calibración y modelado.

Actualmente, el uso masivo de IMUs en estudios de biomecánica, principalmente deportiva y clínica, brinda una gran variedad de sistemas comercializados basados en esta tecnología, diversificándose sus aplicaciones. Algunas de estas aplicaciones son: la monitorización de actividades deportivas, tales como marcha, carrera, ciclismo, o esquí; la monitorización de sujetos patológicos para la prevención patológica, rehabilitación y diseño de prótesis; y otras áreas de impacto de la captura del movimiento.

Sin embargo, el futuro crecimiento de este tipo de tecnologías, además de los beneficios potenciales que ofrece, depara varios retos a afrontar. Si el desarrollo de IMUs se tornara desmesurado, podría abrirse a la implementación inexperta y empleo poco preciso en aplicaciones, así como la aparición de problemáticas relacionadas con la privacidad respecto al uso y explotación de los datos recolectados, cada vez más abundantes.

Entre otros sensores portátiles que se emplean en la monitorización de actividad humana se incluyen los de medición de la temperatura de la piel [22], el ritmo cardíaco [23], la humedad [24] y la frecuencia respiratoria [19]. Además, otros sensores binarios más pequeños han sido propuestos [25, 26] para ser integrados en las prendas de “ropa inteligente” (*smart textiles*) [27].

Para la obtención del conjunto de datos público utilizado en este trabajo, es decir, el PAMAP2 [28] se hizo uso de tres IMUs, particularmente, tres Colibri Wireless de Trivisio [29]. El Colibri Wireless consta de tres sensores del estado del arte de distinto tipo, cada uno con mediciones para tres ejes: un acelerómetro MEMS, un giroscopio MEMS y un magnetómetro AMR (con magnetorresistencia anisotrópica). Además,

cuenta con un sensor de temperatura interno que ayuda a eliminar la influencia de la misma en los demás sensores, y todos tienen una frecuencia de muestreo de 100 Hz. El Colibri Wireless puede emplearse para múltiples fines: dispositivos para juegos y realidad virtual, control de movimiento con interfaz hombre-máquina, dispositivos personales de navegación, maquinaria y robótica, y rehabilitación.

2.1.2 Reconocimiento de actividades

Entre las dos y tres últimas décadas, el estudio dedicado al reconocimiento de actividades humanas (*human activity recognition, HAR*), cuyo objetivo es la identificación de acciones e intenciones de uno o varios agentes mediante sistemas informáticos a partir de una serie de observaciones realizadas sobre los agentes y las condiciones ambientales, ha experimentado un crecimiento notable, creciendo también así los desafíos de diseño, implementación y evaluación de los sistemas a desarrollar.

El reconocimiento automático de actividades requiere trabajar con cantidades de datos con un gran número de dimensiones y caracterizados por una extensa variabilidad, en contextos de la vida diaria, libres de restricciones, por lo que se presenta como un campo de investigación atractivo para el desarrollo de técnicas de aprendizaje automático y la aplicación de sensores portados por el individuo, tales como IMUs. Este tipo de sensores amplía las expectativas de las aplicaciones del reconocimiento de actividades, prescindiendo de ambientes simulados instrumentalmente y ofreciendo un punto de vista subjetivo donde y cuando se requiera.

Por estos motivos, así como por su afinidad con otros campos como la medicina, la interacción entre humano y ordenador, o la sociología, esta disciplina fomenta y sirve de refuerzo a numerosas áreas del mundo real, como el sector industrial [31, 32], los deportes y el entretenimiento [33, 34, 35], y el cuidado de la salud. Algunos dominios en los que se aplica el reconocimiento de actividades son:

- ❖ Cuidado de personas enfermas o inválidas, monitorizando las actividades de la vida diaria (*activities of daily living, ADL*) para poder proporcionar asistencia y rehabilitación [36].
- ❖ Promover una vida saludable y el envejecimiento activo, combatiendo así malos hábitos como la inactividad física.

- ❖ Psiquiatría y psicología, con estudios sobre correlación entre actividades y diferentes trastornos psicológicos.
- ❖ Logística y seguridad, con la monitorización de los trabajadores o de muchedumbres.
- ❖ Soporte a la memoria, mediante la gestión de diarios o registros de manera automática.

DESAFÍOS DE LA MATERIA

Sin embargo, la variedad y la amplitud propias del reconocimiento de actividades también suponen la continua aparición de retos y aspectos a depurar. Por un lado, las actividades humanas no presentan una definición, lenguaje o estructura comunes para establecer un planteamiento común y claro del problema. Por otro lado, la diversidad por naturaleza de las actividades humanas fuerza la utilización de múltiples sensores con distintas características y capacidades. Y, por último, para conocer la calidad de un sistema es necesaria su evaluación según unas métricas específicas para la aplicación destinada.

De este modo, el avance en la investigación sobre el reconocimiento de actividades afronta la resolución de varios aspectos problemáticos, algunos de ellos también existentes en el área del reconocimiento de patrones:

- Variabilidad intra-clase. Implica que varios individuos pueden realizar de manera distinta la misma clase, siendo el caso de la variabilidad inter-sujeto. En el caso de la variabilidad intra-sujeto, hay que tener en cuenta que un mismo individuo también puede llevar a cabo diferentes ejecuciones de una misma actividad. Esto se debe a factores como el estrés, la fatiga, el estado emocional o el ambiente en el que se realiza la actividad.
- Similitud inter-clase. Al contrario que la cuestión anterior, los datos obtenidos por los sensores para diferentes actividades pueden presentar características similares.
- El problema de la clase *null*. En relación con la similitud inter-clase, datos de actividades que resultan irrelevantes para un problema concreto pueden confundirse con actividades relevantes y viceversa, las clases irrelevantes se engloban en el espacio, frecuentemente desconocido y ambiguo, que responde a la clase *null*.

Además, el reconocimiento de actividades en particular, se ve afectado por otros inconvenientes:

- Definición y diversidad de las actividades físicas. La actividad humana es muy compleja y diversa y, como ocurre con la variabilidad intra-clase y la similitud inter-clase, la manera de llevar a cabo una actividad es personal, por tanto, resulta complicado desarrollar un diseño del sistema, con una definición clara de las actividades, que aspire a ser general. En este aspecto, el uso de jerarquías entra en juego permitiendo el reconocimiento de diferentes niveles de actividades, más o menos específicas [37]. Así, se ha logrado realizar buenos experimentos en el reconocimiento de actividades simples, quedando un margen de mejora mucho más amplio en el caso del reconocimiento de actividades complejas. Además, por la naturaleza de las actividades humanas, se derivan otros retos particulares, tales como: reconocer actividades simultáneas [43], reconocer actividades intercaladas [43], lidiar con la ambigüedad en la interpretación, o lidiar con la multiplicidad de agentes en el entorno, reconociendo las actividades realizadas en paralelo o como grupo.
- Balance entre clases. En la mayoría de situaciones reales, diferentes actividades no suceden con la misma frecuencia, dificultando el aprendizaje de aquéllas con menor frecuencia y con una aparición menos significativa en los datos.
- Etiquetado del *ground truth* (información real recogida). Para el aprendizaje supervisado los datos que se almacenan han de ser etiquetados, bien al mismo tiempo que se recogen [39], bien a posteriori, siendo una tarea tediosa pero que debe realizarse con rigor.
- Adquisición de datos y diseño de experimentos. No sólo no resulta sencilla la obtención de los datos experimentales del mundo real, con los costes de, entre otros, instrumentos y participantes, y el tiempo de preparación y ejecución, sino que para poder comparar diferentes sistemas, se ha vuelto imprescindible la utilización de conjuntos de datos estándar en la evaluación de los mismos.

Desde el punto de vista de la aplicación práctica, se hace frente a dos contrariedades adicionales:

- Variabilidad en las características de los sensores. Ésta puede deberse a causas internas, como errores de hardware, fallos o desajuste de los sensores, y externas, como cambios de las

propiedades del entorno, por ejemplo, de la temperatura, o la utilización incorrecta de los dispositivos [40, 41].

- Compromiso entre las cuestiones de diseño. Además, el diseño debe coordinar los puntos mencionados ligados al rendimiento del modelo, con el aprovechamiento de las capacidades del sistema [42]. La relevancia otorgada a cada aspecto dependerá de los recursos disponibles y el problema específico a tratar.

TIPOS DE RECONOCIMIENTO DE ACTIVIDADES

Dentro del reconocimiento de actividades, se pueden diferenciar varios planteamientos según determinados factores como el modo de empleo de los sensores o el tipo de las actividades que se pretende identificar.

Los sensores pueden utilizarse de manera externa a los agentes, situándose en puntos determinados del entorno, o incorporados a los agentes, mediante dispositivos portátiles o *wearables*.

Los sistemas que cuentan con sensorización **externa**, son capaces de reconocer actividades bastante complejas mediante la colocación de sensores en ciertos objetos determinados del ambiente donde se realizan las acciones y con los que pueden interactuar las personas. Sin embargo, de la localización de los sensores depende la cantidad de actividades que el sistema pueda llegar a reconocer, estando éstas limitadas y quedando otras fuera del alcance del mismo. A su vez, la instalación y mantenimiento de los sensores externos suele ser costosa. Una aplicación típica de la sensorización externa son las casas inteligentes [44, 45].

Una variedad de sensores utilizados de manera externa son las cámaras, siendo el foco de numerosos estudios [46-48], con muchas aplicaciones, como la interacción entre humano y ordenador, el diseño de interfaces, el aprendizaje robótico, o la vigilancia, entre otros. A pesar de haberse visto reforzado con la reciente introducción de las cámaras de profundidad que permiten generar esqueletos modelo de los humanos [49], el reconocimiento de actividades con el uso de cámaras implica algunos asuntos delicados como: la falta de privacidad, la restricción del individuo a permanecer dentro del perímetro captado por cámara y la complejidad de las técnicas de procesamiento de vídeo. Las cámaras también pueden emplearse como sensores incorporados, aunque se hace difícil la toma de

imágenes de la actividad que realiza el individuo al completo, siendo más factible una perspectiva de lo que ve el individuo en primera persona.

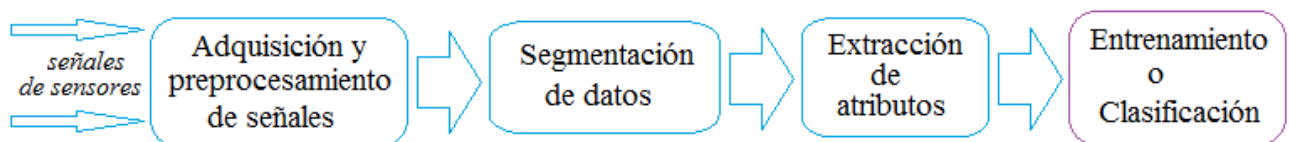
A consecuencia de los impedimentos de los anteriores procedimientos, se impulsa el uso de los **wearables**, que realizan mediciones sobre el movimiento del sujeto, los parámetros del entorno o variables fisiológicas de la persona. (Véase [2.1.1 Monitorización](#))

Finalmente, en función de cómo se desarrollen las actividades, existen sistemas de reconocimiento de actividades de un **único usuario** [50, 51], de **varios usuarios** [52, 53] y **grupales** [54, 55, 56].

CADENA DE RECONOCIMIENTO DE ACTIVIDADES

El reconocimiento de actividades está caracterizado por obedecer a una metodología establecida de manera general: la cadena de reconocimiento de actividades (*activity recognition chain*, ARC), una secuencia de técnicas tanto de procesamiento de señales como de reconocimiento de patrones y aprendizaje automático. La ARC está compuesta de las etapas de adquisición y preprocesamiento de datos de los sensores, segmentación de los mismos, extracción de atributos y entrenamiento o clasificación.

Ilustración 1: Cadena de Reconocimiento de Actividades



Las señales recogidas en los experimentos mediante los sensores conforman la entrada de la ARC, y deben filtrarse teniendo en cuenta que los sensores pueden tener una frecuencia de muestreo distinta o producir señales defectuosas por muchos motivos. En esta etapa, los datos recogidos son preprocesados para sincronizar las señales de los distintos sensores, eliminar la información errónea o inútil y así, preparar los datos,

de manera genérica a todos los sujetos participantes, para la extracción de atributos.

A continuación, las instancias ya preprocesadas son divididas en segmentos que puedan contener información relevante sobre las actividades realizadas. Estas secciones no son triviales de definir y existen varios métodos diferentes para esta tarea, que incluyen desde el uso de los propios sensores de diferentes modalidades hasta segmentación basada en la intensidad de las actividades. Pero la técnica más utilizada en los sistemas de clasificación de actividades es la segmentación de las señales de los sensores en ventanas temporales. Después, cada ventana es tratada como una instancia distinta por el algoritmo de clasificación que se aplique, habiendo que segmentar en ventanas los nuevos datos que se vayan recibiendo en tiempo real. La segmentación en ventanas puede realizarse en base a tres criterios principalmente: las ventanas deslizantes, que dividen la señal en secciones de tamaño fijo, van seguidas o incluso se solapan, y, al no requerir un preprocesamiento extra de los datos en tiempo real, constituyen el método empleado por la mayoría de estudios de clasificación de actividades; las ventanas definidas por eventos, que, como responden a unos eventos especificados previamente en los datos y que no se distribuyen necesariamente de manera uniforme, son de tamaño variable; y las ventanas definidas por actividades, que dividen los datos de acuerdo a los momentos predeterminados como cambios entre actividades, correspondiendo cada ventana a una ejecución de una actividad.

El siguiente paso del proceso consiste en la extracción de atributos, que consta de la generación de atributos de actividades a partir de los datos segmentados y la posterior selección de los atributos más relevantes para la tarea de clasificación en cuestión.

La generación de atributos consiste en la extracción de características relativas a las actividades desde las señales de los sensores, y puede dar lugar a una variedad de atributos distintos que van desde los basados en las propias señales, hasta de múltiples niveles, pasando por los basados en eventos. Los diversos métodos de generación de atributos hasta ahora utilizados en estudios anteriores, pueden agruparse en las siguientes categorías, correspondientes al origen de los atributos obtenidos:

- Atributos heurísticos, es decir, derivados del entendimiento fundamental y normalmente intuitivo de la relación existente entre una postura o movimiento y la señal característica que produce. Es una práctica útil para diferenciar entre posturas estáticas [58, 59, 60] e identificar transiciones entre las mismas [61], así como para distinguir las actividades estáticas de las dinámicas [38] y averiguar el nivel de intensidad de actividad física.

- Atributos de tiempo, frecuentemente son medidas estadísticas generados directamente de los datos de los sensores recogidos en una ventana, por ejemplo, media, mediana, varianza, desviación, curtosis [62, 63] y rango intercuartílico [64], u otras medidas de la correlación entre distintos acelerómetros para mejorar la efectividad de la clasificación [65, 63].
- Atributos de frecuencia, originados tras la transformación de las ventanas de datos, generalmente mediante la transformada rápida de Fourier (*fast Fourier transform, FFT*), corresponden a un conjunto de coeficientes que representan las amplitudes de los componentes de frecuencia de las señales y la distribución de la energía de las mismas. Pueden incluir una sola medida, como la mediana de la frecuencia [66], un subconjunto de coeficientes [67, 68], o combinaciones de varios, como la energía espectral [69, 70] y la entropía de frecuencia [71].
- Análisis *wavelet*, se utiliza para agregar las características de tiempo y de frecuencia, produciendo atributos que informan sobre cambios temporales que se producen en las propiedades de frecuencia, pudiendo analizarse las señales no estacionarias. La transformada *wavelet* discreta (*discrete wavelet transform, DWT*) ha sido utilizada en tres clases de problemas de reconocimiento de actividades: acentuación de las señales de sensores [60, 61, 72], identificación de puntos de transición entre actividades [73, 74] y generación de atributos de tiempo y frecuencia [73, 74, 75, 76, 77, 78, 68].

Una vez convertidas las señales en atributos característicos de una actividad, se realiza una selección de los atributos de mayor relevancia, determinada por la capacidad de discriminar unas actividades y otras [79], es decir, la claridad con la que un atributo separa las instancias en las distintas clases. Además, debe tenerse en cuenta que una cantidad mayor de atributos supone un aumento de los datos necesarios, y por tanto del coste computacional, para el entrenamiento del modelo de clasificación. El mecanismo de selección debe encontrar el equilibrio entre minimizar el número de atributos y obtener la eficiencia deseada, y se puede obtener automáticamente [81] o basándose en conocimiento experto. Algunos ejemplos concretos de métodos utilizados son: el análisis estadístico y visual [19], la selección basada en la correlación [64], la selección según la tasa de acierto para cada subconjunto de atributos [80] y la comparación de distintos atributos y coeficientes estadísticos junto con la aplicación de clustering con k-medias sobre las instancias dentro del espacio de atributos [69].

Una posible alternativa al descarte de ciertos atributos, es la combinación de varios de ellos para formar otros nuevos que reduzcan la dimensión de cada instancia, e incluso aumenten el rendimiento del clasificador. El análisis de componentes principales (*principal component analysis, PCA*) es uno de los métodos de reducción más comunes [82, 83, 84], del que se deriva el análisis de componentes independientes (*independent component analysis, ICA*), y difiere algo más del enfoque con algoritmos basados en múltiples *eigenspaces* [85].

Por último, la ARC dispone de dos modos posibles en su etapa final: el modo de entrenamiento, correspondiente a la fase de entrenamiento o aprendizaje de un algoritmo de clasificación utilizando los datos con los atributos extraídos, y el modo de clasificación, correspondiente a la fase de test o de clasificación de instancias desconocidas, preprocesadas y con los atributos apropiados, para medir el acierto del clasificador producido. La elección del algoritmo de clasificación a utilizar en esta etapa debe hacerse considerando, tanto la minimización de la complejidad y de la exigencia de memoria, como la maximización de la eficacia en el reconocimiento de actividades. En el caso de utilizarse varios clasificadores, la salida de todos debe fusionarse en una única decisión, al inicio o al final de la ARC.

Como se ha mencionado, la evaluación del sistema forma parte de esta etapa de la ARC, y la función objetivo a optimizar dependerá del problema o aplicación concreta de reconocimiento de actividades. Algunas de las métricas utilizadas con mayor asiduidad son la tasa de acierto (*accuracy*), la precisión (*precision*), la cobertura (*recall*), la matriz de confusión (*confusion matrix*), las puntuaciones de F (*F-scores*), la curva característica operativa del receptor (*receiver operating characteristic, ROC, curve*) y de cobertura y precisión (*recall-precision, RP, curve*). El rendimiento de un sistema de reconocimiento de actividades depende de varios aspectos, como la colección de actividades, la calidad del conjunto de entrenamiento, el método de extracción de atributos y el algoritmo de aprendizaje.

Para obtener una evaluación fiable de la robustez del modelo, respecto a su capacidad de generalización en situaciones nuevas, es conveniente realizar diversos experimentos con distintas configuraciones de los conjuntos de entrenamiento y de test. Teniendo en cuenta que la adquisición de datos puede haberse llevado a cabo con varios sujetos, días y ejecuciones de las actividades, la validación cruzada estándar no es suficiente y surgen nuevas adaptaciones: la validación cruzada “dejando un sujeto fuera” (*leave-one-subject-out*), toma los datos de un único sujeto como conjunto de test y utiliza el resto como conjunto de entrenamiento, una vez por cada sujeto, para valorar el sistema de reconocimiento independientemente del sujeto; la validación cruzada “dejando un día fuera” (*leave-one-day-out*), con el mismo proceso pero con cada día; y la

validación cruzada “dejando una ejecución fuera” (*leave-one-run-out*), con los datos de las distintas ejecuciones de un solo individuo. También, puede ser interesante la evaluación específica del sujeto, es decir, ver cómo rinde el sistema en la clasificación de actividades dentro de un mismo individuo, para lo que suele utilizarse la validación cruzada estándar. Finalmente, las métricas de evaluación totales se calculan haciendo la media de los resultados obtenidos para todas las repeticiones correspondientes a las combinaciones de los conjuntos de test y entrenamiento.

A pesar de todas las posibilidades existentes para hacer evaluaciones rigurosas del rendimiento de los sistemas de reconocimiento de actividades, por lo general, los conjuntos de datos empleados y las configuraciones de los experimentos llevados a cabo varían de una investigación a otra, haciéndose muy difícil la realización de comparaciones objetivas entre los sistemas propuestos en estudios distintos.

ASPECTOS DE DISEÑO

Una vez expuesto el procedimiento de los sistemas de reconocimiento de actividades, cabe añadir algunos aspectos adicionales a tener en cuenta en el diseño de los mismos, que se describen a continuación.

Primero, las propiedades que se recojan con los datos dependen de los sensores que se empleen, en particular, con sensores *wearable* se puede percibir cuatro tipos principales de atributos para el reconocimiento de actividades físicas habituales (andar, correr, tumbarse, etc.): los atributos de ambiente, con información sobre el entorno de la persona; de aceleración, posiblemente los más utilizados para el reconocimiento de actividades; de localización, mediante el GPS; y señales sobre el estado fisiológico y vital del individuo (frecuencia cardíaca y respiratoria, temperatura de la piel, conductividad de la piel, etc.) [86].

Continuando en el contexto del uso de *wearables*, el mantenimiento de su carácter práctico es fundamental. Para que un sistema pueda aplicarse al reconocimiento de actividades, no puede excederse en el uso de sensores, y es que, a pesar de aumentar sus fuentes de información, el sistema puede resultar intrusivo, caro e incómodo para el usuario.

Otro factor importante respecto a la adquisición de datos, es el protocolo seguido por los individuos participantes, así como la cantidad de

los mismos y sus características físicas. Para que un estudio sea lo más realista posible, debe ser genérico, involucrando a un extenso y diverso grupo de personas, con diferente género, edad, altura, peso y condiciones de salud. De esta manera se favorece que el clasificador elaborado sea capaz de aplicar su aprendizaje con éxito de cara a usuarios desconocidos.

Sin embargo, existe un debate relacionado con este último punto. Mientras unos autores defienden el desarrollo de sistemas flexibles con capacidad de generalización, independientes del usuario [87], otros consideran que la mejor manera de resolver el problema de la diversidad en los usuarios y en su forma de realizar las actividades, es adaptar el modelo de clasificación a cada individuo específicamente [88], teniendo que repetirse el entrenamiento con cada nuevo usuario.

En definitiva, el diseño global de un sistema de reconocimiento de actividades debe responder a la aplicación final, la cual dictamina las actividades a reconocer, el acierto o precisión de identificación de actividades que se exige y los sensores más apropiados para su uso práctico.

MÉTODOS DE APRENDIZAJE

Con respecto al acierto requerido, su consecución, así como la superación de muchos de los desafíos planteados anteriormente, dependen del algoritmo que se emplee para la inferencia de conocimiento a partir de los datos. En este trabajo, ya se ha hecho referencia a que un problema de reconocimiento de actividades se asume como una tarea de clasificación. Es por esto, que muchas técnicas de aprendizaje automático sean, no las únicas, pero de las más aplicadas por este tipo de sistemas, debiendo escogerse según su adecuación al problema en cuestión, que depende diversos aspectos como la facilidad de desarrollo y la capacidad de responder en tiempo real. Se dice de un sistema que es *online* si realiza la clasificación de instancias nuevas en tiempo real, y *offline* si realiza la clasificación después de almacenarlas.

Para concluir el estudio referente al estado del arte del reconocimiento de actividades, se presenta un resumen de los métodos empleados en investigaciones previas, agrupados de acuerdo al tipo de aprendizaje que hagan: supervisado, con un conjunto de datos completamente etiquetado, o no supervisado, cuando los datos no incluyen la clase.

La mayoría de los sistemas funcionan con aprendizaje supervisado, ya que la salida generada debería ser la etiqueta de una actividad humana. A su vez, diferenciar las actividades en un contexto absolutamente no supervisado puede ser una ardua tarea, y por ello, existen algunos sistemas considerados semisupervisados, que funcionan conociendo las etiquetas de parte de los datos.

Aprendizaje supervisado

El aprendizaje supervisado, y concretamente la clasificación, ha brindado una variedad de algoritmos para el reconocimiento de actividades humanas, algunos de los cuales se exponen a continuación junto con estudios que han hecho uso de los mismos.

La **clasificación basada en umbrales** (*threshold-based classification*), en la que simplemente un atributo es comparado con un umbral para determinar si se trata de una actividad u otra. Este método se ha aplicado con éxito en la diferenciación entre posturas estáticas, como estar de pie, sentado o tumbado, mediante información angular obtenida con acelerómetros situados en la cadera o el tronco [89, 90, 91], en las extremidades inferiores [92, 93, 90, 94] y en el pecho [58, 61]. La información angular de acelerómetros y de giroscopios también ha dado buenos resultados en la identificación de transiciones entre posturas [60, 61]. Para distinguir las posturas estáticas de las posturas dinámicas, es común el empleo de un atributo que mida la variación de la señal de aceleración [38, 59, 95]. En cuanto a la clasificación de actividades dinámicas, pocos investigadores han logrado buen rendimiento con este método [96, 97, 74, 75, 73].

La **clasificación jerárquica**, consiste en una estructura de decisión binaria que se construye manualmente, consta de una secuencia de nodos y, por su carácter intuitivo y fácilmente actualizable en caso de querer añadir actividades [98], se ha utilizado en múltiples investigaciones [99, 100, 101, 19, 102]. A pesar de que su desarrollo puede ser largo y tedioso, requiere un esfuerzo computacional mínimo, siendo adecuado para las aplicaciones en tiempo real.

Los **árboles de decisión**, también representan un modelo jerárquico de clasificación, con nodos correspondientes a los atributos y hojas por cada clase. En cambio, su construcción se automatiza, ganando velocidad y disminuyendo la dependencia de que intervenga el usuario. Una gran variedad de problemas se han resuelto haciendo uso de árboles de decisión [102, 19, 51, 71], y si se requiere, pueden aplicarse online, una vez desarrollados [64]. Tanto en el caso de los árboles de decisión como el de

la clasificación jerárquica, se puede disponer de una representación gráfica para el entendimiento e interpretación de las reglas generadas.

El **aprendizaje basado en instancias** (*instance-based learning, IBL*), identifica una nueva instancia según la clase de la instancia de entrenamiento más cercana, es decir, similar. El método más relevante en el reconocimiento de actividades es el de los “k vecinos más cercanos” (*k nearest-neighbors, k-NN*), que sigue el mismo proceso pero son las k instancias de entrenamiento más cercanas las que deciden, por mayoría, la clase de la nueva instancia. Su desarrollo es rápido, es muy versátil y admite un amplio rango de actividades, sin embargo, la ejecución online puede ser demasiado lenta a causa de todos los cálculos de distancia necesarios. La primera vez que este algoritmo fue empleado en la clasificación de actividades, fue en 1999 por Foerster et al [4], ampliándolo en el año 2000 [66]. Posteriormente, se han utilizado otros enfoques similares [103], ha sido contrastado con otros métodos [71, 64] y ha sido integrado en la comparación de atributos de actividades [69, 67].

Las **máquinas de vectores de soporte** (*support vector machines, SVMs*), [104, 105] constituyen un método de aprendizaje automático bastante popular basado en encontrar los hiperplanos que separen las clases con el máximo margen entre las instancias de cada clase. Respecto al reconocimiento de actividades, hay algunos estudios que las utilizan, y aunque su representación no sea intuitiva, han demostrado tener potencial en este tipo de problemas [106, 107]. Las SVMs son fiables en la clasificación de conjuntos complicados y con ruido, pero su entrenamiento puede ser demasiado lento.

Las **redes de neuronas artificiales** (*artificial neural networks, ANNs*), actúan como una función matemática flexible que representa relaciones complejas entre sus entradas, variables independientes, y salidas, variables dependientes. Al ser una técnica flexible, las ANNs son potentes y se pueden emplear en una amplia variedad de problemas de clasificación, así como en la predicción de parámetros funcionales [65, 108, 109, 63]. Una de las redes de neuronas más comunes es el perceptrón multicapa (*multilayer perceptron, MLP*) [110, 111], y se ha aplicado con buenas tasas de aciertos en numerosos estudios de reconocimiento de actividades [112, 62, 114, 78], siendo a su vez comparado con otras resoluciones [102, 19, 80]. Por otro lado, el entrenamiento de las ANNs puede ser lento, y algunas resultan difíciles de implementar.

Los **clasificadores bayesianos**, calculan las probabilidades de las instancias desconocidas de pertenecer a cada clase a partir de las probabilidades condicionales estimadas del conjunto de entrenamiento. Son simples de implementar y de rápida ejecución pero dan por sentada la independencia entre atributos, y aún con buenos resultados para

problemas con pocos sujetos [64, 51], en la mayoría de los casos su rendimiento es más bajo que el de otros algoritmos [71, 106, 114, 115]. Los modelos de mezclas gaussianas (*gaussian mixture models*, *GMMs*) son mejores que los clasificadores con Naive Bayes [116].

Las **cadenas de Markov**, son modelos gráficos que contienen información sobre la probabilidad de transición entre diferentes estados de actividad. Los modelos ocultos de Markov (*hidden Markov models*, *HMMs*) son una extensión de las cadenas de Markov simples, y son muy eficaces en la identificación de una secuencia de actividades a partir de una secuencia de instancias. La clasificación resultante no depende sólo de las instancias observadas, sino también de la probabilidad de una transición desde una actividad anterior. Se ha demostrado que los HMMs son buenos como clasificadores aislados [117] y para complementar y mejorar el rendimiento de otros clasificadores [118, 24, 119].

La **lógica borrosa**, proviene de la teoría de conjuntos borrosos y asocia unas entradas a una o varias salidas mediante declaraciones condicionales o reglas. Dado que el razonamiento que usa permite conceptos imprecisos y aproximaciones, abre un frente esperanzador de cara al reconocimiento de actividades humanas. Aún así, presenta algunas dificultades como la construcción de las funciones de pertenencia y la interpretación única de las reglas borrosas. Pocos estudios previos han logrado una buena precisión de clasificación mediante técnicas de lógica borrosa [120, 121].

Los **conjuntos de clasificadores**, combinan la salida generada por distintos clasificadores para obtener un mejor rendimiento. Para ello, se fusionan las distintas clasificaciones mediante métodos como *bagging*, *boosting*, *stacking* o varios tipos de voto mayoritario, complementándose unas decisiones con otras. De esta manera, se puede obtener un clasificador robusto a partir de clasificadores simples, aunque cause un mayor coste computacional para entrenar y evaluar todos. Es un enfoque que ha ganado popularidad en la comunidad biomédica y resulta prometedor desde el punto de vista del reconocimiento de actividades, con diversos estudios [51] en los que es frecuente el uso del metaclassificador AdaBoost [118, 24, 25].

Aprendizaje no supervisado

En la práctica, etiquetar los datos puede ser complicado en algunas situaciones, sobre todo cuando la granularidad de las actividades es muy alta o el usuario no está dispuesto a cooperar en el proceso de adquisición de datos. El aprendizaje no supervisado puede usarse para el análisis y la interpretación de los datos de sensores evitando tener que etiquetar cada ventana o instancia de los mismos, así como para contribuir en el desarrollo del aprendizaje supervisado proporcionando entendimiento de la estructura interna de los datos de actividades.

Así, se tiende a la utilización de un aprendizaje más bien semi-supervisado, es decir, contando con las etiquetas de parte de los datos [122, 123, 124, 125, 106] o a la combinación de métodos no supervisados con una capa supervisada. A pesar de la gran utilidad potencial de esta estrategia en el campo del reconocimiento de actividades, pocos estudios la han aplicado. Por ello, no se puede establecer unos algoritmos estándar, pero un ejemplo de las técnicas empleadas en investigaciones anteriores son los **mapas autoorganizados de Kohonen** (Kohonen *self-organizing feature map*, SOM) [126,22].

FUTURO DE LA MATERIA

Los sistemas de reconocimiento desarrollados en trabajos anteriores abordan los asuntos más complejos del campo, así como los retos planteados previamente, con una gran variedad de tácticas. No obstante, hay algunos aspectos que se necesita continuar estudiando en futuras investigaciones [1]:

- ❖ La comparación entre los enfoques implementados en distintos estudios está limitada por la falta de conjuntos de datos de reconocimiento de actividades que se utilicen de manera extendida y común a las diferentes investigaciones. Hay algunos conjuntos de datos publicados por universidades e instituciones [127-131], y para eventos, como el proporcionado en el *Activity Recognition Challenge* de 2011 [132], en el que se invitaba a los investigadores de todo el mundo a participar.
- ❖ Las actividades consideradas en general son bastante simples, algunas siendo las que conforman rutinas o comportamientos. Se enriquecería la correspondencia con el entorno real del

reconocimiento de actividades, si se lograra identificar actividades más complejas a partir de otras simples [37].

- ❖ La concurrencia y solapamiento natural de las actividades apenas se considera en los trabajos realizados, asumiéndose erróneamente que todas las actividades humanas se llevan a cabo por separado [133].
- ❖ La clasificación de atributos, no sólo pudiéndose deducir actividades sino también atributos del usuario.
- ❖ La clasificación sensible al coste [86, 134, 135], es decir, que tenga en cuenta el coste de predecir una actividad en concreto cuando la correcta es otra actividad determinada. Estableciendo estas relaciones según la aplicación específica del sistema, el clasificador escogerá como salida aquella clase que, de estar equivocada, suponga el menor coste. Incluso se podría considerar el coste aumentando la presencia de las actividades con mayor coste por clasificarse mal, fomentando el aprendizaje de éstas por parte del algoritmo.
- ❖ El reconocimiento de actividades colectivas, no necesariamente grupales, para obtener información sobre el comportamiento de distintas poblaciones, demarcadas por zona u otras características, a partir de muestras significativas de las mismas.
- ❖ La predicción de actividades futuras, obteniéndose sistemas que pronostiquen qué va a hacer un usuario. Algunos trabajos previos también incluyen la predicción de rutinas de comportamiento [136].
- ❖ El conflicto de la flexibilidad de los clasificadores podría abordarse con un enfoque alternativo, a medio camino entre una solución general para todos los individuos y una solución adaptada a cada usuario, elaborando clasificadores asociados a grupos específicos o clusters de individuos con características similares.

2.1.3 Trabajos relacionados

En este apartado, se consideran trabajos relacionados aquéllos realizados con el mismo conjunto de datos que el utilizado en este proyecto [131, 151, 28, 145, 147, 148, 149, 150, 152]. Estos sistemas pueden compararse con el aquí propuesto, con mayor fiabilidad que el resto de planteamientos, aún sin una fiabilidad total puesto que la configuración de los métodos de evaluación y las ejecuciones de pruebas varían de un estudio a otro, a pesar de utilizarse la misma fuente de datos.

Se han seleccionado algunos de los trabajos llevados a cabo con el conjunto PAMAP2, para analizar los diversos enfoques y comparar, en la medida de lo posible, los resultados obtenidos con cada uno de ellos.

En el propio estudio que introduce el conjunto de datos (1) [28], de A. Reiss y D. Stricker, se realiza una comparativa de rendimiento entre varios clasificadores de la herramienta Weka: árbol de decisión C4.5, *boosted* C4.5, *bagging* C4.5, Naive Bayes y kNN. Estos algoritmos se aplican a cuatro problemas, tres de los cuales son de reconocimiento de actividades: reconocimiento de 5 actividades principales, reconocimiento de 5 actividades principales, más una categoría que engloba, como una, a otras 7 restantes, y reconocimiento de las 12 actividades. Para la evaluación, se realiza validación cruzada independiente del sujeto y dependiente de cada sujeto. En el caso del reconocimiento de 12 actividades, que es el más cercano al problema que se trata en este trabajo, se consigue una tasa de acierto máxima de 99.69%, con el *boosted* C4.5 para la validación dependiente del sujeto, y de 89.24%, con el kNN para la validación independiente del sujeto.

Otro estudio relacionado, realizado por los mismos autores (2) [147], propone un algoritmo que combina varios clasificadores para el reconocimiento de actividades personalizado. La personalización se basa en el re-entrenamiento de los pesos asignados a los distintos clasificadores, previamente generados con datos de otros sujetos, mediante la adición de datos del sujeto desconocido. La evaluación se realiza para dos problemas: reconocimiento de 6 actividades y reconocimiento de 15 actividades. Según el mecanismo de re-entrenamiento, la validación cruzada es parcialmente dependiente del sujeto. Para el problema de 15 actividades, el más parecido al de este trabajo, se logra un rendimiento, concretamente, una máxima puntuación de F de 74.41%, aproximadamente, con el clasificador AdaBoost.M1.

Siguiendo la misma línea, A. Reiss, D. Stricker y G. Hendeby, en su trabajo sobre el reconocimiento robusto de actividades del día a día (3) [148], realizan una evaluación de diferentes métodos, con validación cruzada dependiente e independiente del sujeto. Los métodos se aplican a diversos problemas: uno de reconocimiento de 6 actividades, y cuatro de reconocimiento de hasta 15 actividades, tratando las 9 adicionales de maneras distintas en cada uno. Para el problema que trata las 15 actividades como actividades distintas separadas, el más afín al presentado en este trabajo, se consigue una tasa de acierto del 99.83% para la validación dependiente de sujeto y del 86.6% para la validación independiente de sujeto, en ambos casos utilizando el clasificador AdaBoost.M1.

Por otro lado, A. Baldominos, Y. Sáez y P. Isasi, centran su estudio en la optimización del conjunto de atributos considerados haciendo uso de algoritmos genéticos (4) [145]. Se comparan los resultados de varios clasificadores de actividades para un conjunto de atributos genérico, frente a los obtenidos para un conjunto de atributos optimizado para cada sujeto mediante un algoritmo genético. En el primer caso, la mejor tasa de acierto la consigue el clasificador *Random Forest*, con un 93.51%, que se incrementa hasta un 98.89% con la optimización de la selección de atributos personalizada, y con el mismo clasificador. Ambos resultados reflejan el rendimiento evaluado con validación cruzada independiente del sujeto.

Por último, el reciente estudio realizado por Bleser et al, sobre la monitorización de actividad física personalizada utilizando sensores *wearable* (5) [150], presenta un nuevo método de clasificación de actividades, llamado ConfAdaBoost.M1, una extensión del algoritmo AdaBoost.M1 basada en valores de confianza en la predicción. Aplicando este algoritmo a un problema con 6 actividades principales y una adicional que engloba el resto de actividades del contexto, se obtiene una tasa de error aproximada del 22.22% en la validación cruzada independiente del sujeto.

Tabla 1: Trabajos relacionados

	LOSO CV	Intra-sujeto CV	Otra
1 (12 actividades)	0,8924 (kNN)	0,9969 (Boosted C4.5)	X
2 (15 actividades)	X	X	74,41% (Ada Boost.M1)
3 (15 actividades)	86,6% (AdaBoost.M1)	99,83% (Ada Boost.M1)	X
4 (12 actividades)	93,51% (Random Forest)	X	X
4 (12 actividades Opt)	98,89% (Opt-Random Forest)	X	X
5 (6+"contexto")	22,22% (ConfAdaBoost.M1)	X	X

2.2 Técnicas

2.2.1 Aprendizaje automático

La sociedad produce ingentes cantidades de datos, de fuentes de todo tipo, que encierran información potencialmente valiosa. Sin embargo, los datos en bruto, sin procesar, no bastan, volviéndose necesaria la aplicación de métodos para la extracción automática de información útil.

El aprendizaje automático es la disciplina de inteligencia artificial que trata el desarrollo de técnicas que resuelvan problemas y, a su vez, tengan la capacidad de modificar su comportamiento de manera automática para lograr un mejor rendimiento o resolver nuevos problemas. Para ello, es necesario adquirir conocimiento basándose en la experiencia, y el análisis inteligente de datos es un gran recurso. En este campo, se pretende elaborar algoritmos que encuentren y describan patrones estructurales presentes en ejemplos, sirviendo como herramienta para ayudar a explicar esos datos y realizar predicciones en nuevas situaciones a partir de ello.

La faceta más práctica del aprendizaje automático es la minería de datos, y ambas están estrechamente relacionadas con la estadística y la búsqueda en bases de datos.

La amplia variedad de aplicaciones prácticas del aprendizaje automático, lo convierten en un dominio bastante atractivo para su estudio, y algunos ejemplos de áreas en las que resulta útil son: la toma de decisiones automática, análisis y reconocimiento de imágenes, pronóstico de consumo, diagnóstico, marketing y ventas, mantenimiento automático de sistemas, software personalizado, etc.

Dentro del aprendizaje automático, existen distintos tipos de tareas según el problema a resolver, con los mismos componentes de entrada generalmente: el concepto a aprender, las instancias o ejemplos individuales del concepto y los atributos o aspectos medibles de una instancia. Además, dependiendo de si se conoce o no la salida esperada correspondiente a la entrada, las tareas son de aprendizaje *supervisado* o *no supervisado*, respectivamente.

Por un lado, se consideran tareas de aprendizaje *supervisado* la clasificación, donde a partir de instancias ya etiquetadas se pretende aprender a etiquetar nuevas instancias, la regresión, una variante en la que la salida es numérica en lugar de una clase discreta, y la asociación, en la

que se trata de averiguar relaciones entre valores de atributos para predecir, no sólo la clase, sino también uno o varios atributos.

Por otro lado, es una tarea de aprendizaje *no supervisado* la agrupación o clustering, donde se buscan los grupos que reúnan instancias similares entre sí.

Así como hay distintas tareas de aprendizaje automático, existen diferentes maneras de describir el conocimiento adquirido en el proceso. No obstante, todas ellas tienen algo en común: en cualquier tarea de aprendizaje automático, es deseable que la descripción de los patrones estructurales aprendidos, es decir, la salida del algoritmo, sea fácilmente comprensible. Es más, el principio de longitud de descripción mínima, MDL (*minimum description length*), defiende que la mejor descripción es aquella con la menor longitud posible, siendo esta última el total de espacio requerido para describir la teoría deducida y los errores de la teoría. En definitiva, se trata de un criterio de diseño que busca un buen equilibrio entre la complejidad de un modelo y su rendimiento.

Otro aspecto relacionado con la complejidad del modelo generado, también importante independientemente del concepto a aprender, es la capacidad de generalización del mismo. Por lo general, cuanto más simple sea la estructura definida, menos riesgo de caer en el sobreajuste (*overfitting*), también llamado sobreentrenamiento. El sobreajuste se refiere a la especialización excesiva de un modelo en los casos concretos del conjunto de entrenamiento, es decir, cuando describe con gran precisión las instancias de aprendizaje pero no se ajusta bien a las instancias externas. Entre las causas de este problema se incluyen el ruido en los ejemplos, la escasez de los mismos, una estructura errónea del modelo, etc.

En este documento, se profundiza en la clasificación y el clustering, puesto que son las tareas aplicadas en el proyecto.

CLASIFICACIÓN

En la clasificación, se dispone de un conjunto de datos con un esquema de entrada que incluye la salida real, concretamente, la clase del ejemplo, y se genera un clasificador a partir de estos datos. El éxito del aprendizaje se puede medir con la tasa de acierto obtenida por el clasificador generado para instancias no utilizadas previamente pero cuya

clase se conoce. Así, el procedimiento apropiado divide los datos disponibles en tres conjuntos: el conjunto de entrenamiento, el conjunto de validación y el conjunto de test.

Durante la fase de aprendizaje, se aplica el algoritmo de clasificación al conjunto de entrenamiento, obteniéndose una descripción del concepto aprendido, es decir, un clasificador. Para la optimización de los parámetros del algoritmo, se utiliza el conjunto de validación. Finalmente, se realiza la evaluación del clasificador con el conjunto de test, que cuanto más grande y diverso sea, más fiable será la estimación de la tasa de error obtenida.

Una vez terminado el proceso completo, el clasificador final puede generarse utilizando todos los datos de los tres conjuntos como conjunto de entrenamiento, ya que, de igual forma que el conjunto de test, generalmente un conjunto de entrenamiento más extenso favorecerá una fase de aprendizaje más efectiva.

Para solventar este conflicto con el mayor aprovechamiento posible de los datos disponibles, principalmente se emplea la validación cruzada con k subconjuntos (*k-fold cross-validation*), también llamada validación cruzada con k iteraciones. Este método consiste en dividir el conjunto de datos en k subconjuntos de igual tamaño y, por turnos, usar un subconjunto para test, y dejar el resto para entrenamiento. Es decir, en total se realiza k veces, siendo la tasa de acierto la media de las obtenidas en cada iteración.

El método estándar se realiza con 10 subconjuntos estratificados, es decir, que idealmente contengan instancias pertenecientes a todas las clases, estando ampliamente demostrado en experimentos y habiendo ciertas pruebas teóricas de ser la mejor configuración para obtener una estimación rigurosa del error [154].

Otra manera particular de aplicar la validación cruzada es “dejar uno fuera” (*leave-one-out cross-validation*), la cual equivale a realizar la de k subconjuntos pero fijando k al número de instancias del conjunto de entrenamiento. Por tanto, se repite el proceso tantas veces como instancias haya y usándose cada vez una instancia como test. Este tipo de validación explota al máximo los datos disponibles y no implica la aleatorización de los mismos, aunque es de alto coste computacional e imposibilita la estratificación.

En el caso de la clasificación, el modelo sería el clasificador generado y se pueden encontrar varias alternativas para representarlo, tales como tablas, reglas y árboles. La tabla es la estructura más simple de

representación, indica qué clase corresponde a ciertos valores de los atributos recogidos en la misma, sin embargo, puede suponer una selección, no tan simple, de los atributos más relevantes. A semejanza de la tabla, las reglas son condiciones representativas de los datos, formadas por el antecedente o precondition referente a unos atributos, y el consecuente o conclusión que se aplica a las instancias que lo cumplan. Hasta aquí, tanto reglas individuales como en conjunto son simples, no obstante, si no se dispone de ninguna información adicional, su interpretación puede complicarse al desconocerse el orden de ejecución o existir conflictos entre diferentes reglas, es decir, conclusiones distintas para una misma instancia. En cambio, los árboles presentan una estructura con una jerarquía definida.

De la idea “divide y vencerás” (*divide-and-conquer*) surge la representación mediante árboles de decisión, en los que cada nodo corresponde a una condición de un atributo y cada hoja, a una clase. Si el atributo es de tipo nominal, es analizado por un único nodo con tantos hijos como valores tenga el atributo. Si, por el contrario, el atributo es de tipo numérico, puede aparecer en varios nodos donde se compara con un valor constante, siguiendo una rama u otra en caso de ser mayor, o menor o igual, respectivamente.

Existen otras variedades dentro de los árboles, como el árbol funcional (*functional tree*), que en sus nodos compara una función de varios atributos, en vez de comparar un solo atributo. Además, hay árboles para predicción numérica, los llamados árboles de regresión, en los que cada hoja corresponde a la salida media de las instancias que la alcanzan. Una versión más sofisticada de los árboles de regresión son los árboles de modelos (*model trees*), cuyas hojas contienen ecuaciones de regresión en lugar de simples valores.

Volviendo a los árboles de decisión, existen múltiples implementaciones pero la más utilizada hoy en día es la del C4.5, una extensión del procedimiento de construcción de árboles de decisión conocido como ID3, ambos algoritmos propuestos por J. Ross Quinlan [162-165]. El ID3 consiste básicamente en: seleccionar un atributo para colocar en la raíz y añadir una rama por cada valor posible, recursivamente, hasta que todas las instancias de ese recorrido sean de la misma clase, la cual se pone como hoja.

Por consiguiente, el árbol se construye de arriba a abajo, y haciendo uso de la Teoría de la Información de C. Shannon [166] para la selección del atributo correspondiente: en cada paso, se obtiene el atributo considerado más útil en base a la ganancia de información que proporciona, que se calcula a partir de la entropía de Shannon.

La ganancia de información de un atributo equivale a la entropía total del sistema una vez añadido y dividido en sus valores el nodo con dicho atributo:

$$Gain(A) = E(T) - E(T,A)$$

Siendo $E(T)$ la entropía del sistema antes de aplicar el atributo y $E(T,A)$, tras aplicarlo. La entropía es comparable a la homogeneidad en la distribución de las clases, y se calcula con las siguientes fórmulas, donde $E(r)$ es la entropía de cada rama del nodo A , $P(r)$ es la proporción de instancias que bajan por esa rama y $p(i)$ es la proporción de instancias de esa rama pertenecientes a la clase i :

$$E(r) = \sum_{i=1}^C -p(i) \log_2 p(i) \quad ; \quad E(T,A) = \sum_{r \in A} P(r)E(r)$$

De esta manera, cada vez se añade el atributo que mejor separa las instancias en las clases existentes, dando lugar a un algoritmo bastante robusto. A pesar de ello, el ID3 no tiene en cuenta aspectos como los atributos numéricos, la existencia de valores desconocidos o ruido en los datos, ni la extracción de reglas a partir de árboles, aspectos que sí están incluidos, entre otros, como mejoras en el C4.5.

Una implementación del algoritmo C4.5, el J48, se encuentra entre los métodos de aprendizaje automático disponibles en la herramienta Weka, utilizada para este trabajo.

Weka (*Waikato environment for Knowledge Analysis*) fue desarrollada en la Universidad de Waikato en Nueva Zelanda, en lenguaje Java, y proporciona una amplia colección de algoritmos de aprendizaje automático, así como diversas herramientas para el preprocesamiento de datos, y otras funcionalidades de soporte para la evaluación estadística de esquemas y la visualización de los datos y resultados obtenidos.

AGRUPACIÓN O CLUSTERING

Las técnicas de clustering se utilizan cuando no hay una clase a predecir y se quiere agrupar las instancias en sus grupos naturales, es decir, se busca la configuración de clusters que asocie las instancias más parecidas entre sí.

Hay distintos tipos de clustering, según la naturaleza de los datos: los grupos pueden ser exclusivos, es decir, que cada instancia forme parte de un único cluster, o pueden solaparse y que una misma instancia esté en varios clusters; otra posibilidad es que cada instancia tenga cierta probabilidad de pertenecer a cada cluster, o puede haber una jerarquía donde los grupos de instancias sean más reducidos y específicos a medida que se baje de nivel.

En el caso de que se pretenda organizar las instancias en un número determinado de clusters exclusivos, se puede aplicar una técnica simple y directa muy utilizada, la clásica *k*-medias (*k-means*). En el algoritmo *k*-medias, la *k* corresponde al número de clusters a definir y, por tanto, al número de puntos del espacio de soluciones que se escogen como centros iniciales de los clusters. Una vez escogidos aleatoriamente los centros iniciales, se asignan todas las instancias al cluster cuyo centro es más cercano, según su distancia Euclídea. Luego, se calculan la media de las instancias de cada cluster, obteniéndose los centroides de cada uno, y se repite el proceso hasta que los centroides, tomados como centros de los clusters, no varíen.

La estabilización de los centroides indica que se ha encontrado un mínimo local, lo que significa que el procedimiento no garantiza la localización del mínimo global. La calidad de la solución encontrada depende en gran medida de los centros iniciales elegidos, llamados semillas y, para aumentar la probabilidad de encontrar el óptimo global, se puede ejecutar el algoritmo varias veces con diferentes semillas y quedarse con la mejor solución. Además, existe una evolución del método, el *k*-medias++, que en vez de hacer la selección de todas las semillas aleatoriamente: elige la primera semilla aleatoriamente de una distribución uniforme, y cada semilla restante, con una probabilidad proporcional a su distancia a la semilla ya elegida más cercana. Este procedimiento consigue una mayor precisión de clustering y acelera la ejecución del mismo, a pesar de ralentizar la fase de definición de semillas.

Tanto el *k*-medias como el *k*-medias++, son métodos iterativos basados en la distancia, útiles únicamente cuando se conoce el número de clusters a definir. En caso de desconocer la cantidad óptima, y se tanteen diferentes valores de *k*, es conveniente aplicar el ya mencionado criterio MDL, penalizando las configuraciones con mayor número de clusters.

Existen otros métodos orientados a los distintos tipos de clustering: algoritmos aglomerativos para el clustering jerárquico, procedimientos basados en la utilidad de categoría (*category utility*) del clustering incremental, modelos estadísticos de mezclas finitas (*finite mixtures*) como el algoritmo EM en el clustering probabilístico y el AutoClass en el clustering bayesiano.

2.2.2 Computación evolutiva

Una de las ideas más influyentes en el pensamiento humano desde los filósofos griegos de la Antigüedad, y ya aceptada, aunque no probada, como teoría científica en el siglo XVIII, es la evolución de los seres vivos. Los sucesivos trabajos relativos al tema, como el de Jean-Baptiste Lamarck a principios del siglo XIX, los escritos sobre el evolucionismo de A. R. Wallace [167] y Charles Darwin [168, 169], recopilados en 1858 [170], y, finalmente, el libro del propio Darwin en 1859, *On the origin of species*, [171] sobre la selección natural, hicieron que las ciencias de la vida se centraran en la teoría de la evolución.

Del trabajo de Darwin se derivan varios principios:

- El mundo no es constante, ni fue creado recientemente, ni está en perpetuo ciclo sino en continuo cambio, así como los organismos se van transformando a lo largo del tiempo, es decir, evolucionan.
- Igual que cada grupo de seres vivos desciende de un mismo ancestro, todos ellos provienen de un origen común de la vida en la Tierra.
- Las especies se multiplican, bien reproduciéndose o bien mediante la evolución en nuevas especies.
- Estos cambios evolutivos son graduales en poblaciones, no se producen nuevas especies a partir de individuos de manera repentina.
- Según la selección natural, dentro de cada generación, sobreviven los individuos con las variaciones genéticas mejor adaptadas, y las siguientes generaciones las heredan.

A estos antecedentes hay que añadir la influencia de los avances conseguidos en el área de la genética. Son fundamentales las leyes de Mendel sobre la herencia de caracteres según el cruce de los mismos, elaboradas a partir de sus experimentos realizados entre 1857 y 1868 [172], pero no aceptadas hasta más de 30 años después. A las que, posteriormente, siguieron los diversos descubrimientos referentes al ADN y al ARN. Como consecuencia, se introdujeron, entre otros, los siguientes conceptos:

- Gen: se considera aquello que produzca una característica concreta.
- Genotipo: es la secuencia de bases enumeradas una detrás de otra, el código genético.

- Fenotipo: la característica concreta producida por un determinado gen.

En todas estas ideas, hay ciertos componentes que inspiran modelos potenciales de cómputo, que ingenieros y científicos han interpretado y formalizado dando lugar a la computación evolutiva.

Ya en los años sesenta y setenta, se empezaban a usar los términos “programación evolutiva” con las máquinas de estados finitos de Fogel y sus diversos estudios [173-179], “estrategias evolutivas” con los trabajos de Rechenberg y Schwefel [180-183] y “algoritmo genético” con la población de cadenas de dígitos binarios de Holland (1975) [184]. Siendo John Holland uno de los pioneros más influyentes, considerado padre de la disciplina, gracias a su obra *“Adaptation in Natural and Artificial Systems”* [184] que permite comprender la potencia computacional de esta tecnología.

Sin embargo, no es hasta la última década del siglo pasado cuando el campo de la computación evolutiva florece: la aplicación de los algoritmos genéticos se extiende a una gama muy variada de problemas y nacen estrategias nuevas, como la programación genética con la población de árboles de datos de Koza (1992) [185]. Crecimiento que se vio favorecido por el nuevo concepto de inteligencia artificial desarrollado entre los años ochenta y noventa por, entre otros, Marvin Minsky [186], Hans Moravec [187, 188] y el investigador del MIT (*Massachusetts Institute of Technology*) Rodney Brooks, en su artículo sobre la *Nouvelle AI*, “Elephants don’t play chess” [189]. Esta vertiente de la inteligencia artificial, llamada de forma más genérica inteligencia computacional, se separa de la línea simbólica tradicional, y se orienta a la consecución de módulos de comportamiento simples, tales como neuronas, partículas u organismos, de cuya cooperación e interacción como colectivo puedan emerger comportamientos complejos.

Ya entrado el siglo XXI, denominado siglo de la biología, la computación evolutiva pasa a ser una de las estrategias representativas en el área de la inteligencia computacional, gracias en gran medida a la potencia de sus técnicas.

Esta potencia radica en que la computación evolutiva puede prescindir de analizar cada problema concreto, encontrar una solución y programarla, pues únicamente debe definir una representación de la solución y del problema para resolverlo en tiempo razonable con un

algoritmo general. Al igual que en otras técnicas englobadas en el campo de *soft computing*, como redes de neuronas, optimización con enjambres de partículas y lógica difusa, no es necesario especificar “cómo hay que buscar” sino solamente “qué hay que buscar”.

Ligadas a la computación evolutiva, aparecen otras técnicas de inspiración biológica, basadas en comportamientos presentes en la naturaleza, principalmente aquellas estrategias con una analogía social, las empleadas por poblaciones de individuos que interactúan para lograr un objetivo, no necesariamente común. Es el caso de la optimización con enjambres de partículas, la optimización con colonias de hormigas o la coevolución.

La optimización con enjambres de partículas surge de la observación del comportamiento social de las bandadas de pájaros o los bancos de peces, iniciada por R. Eberhart y J. Kennedy en 1995 [193-195]. En su origen, el objetivo era simular gráficamente la coreografía de estas especies, en cambio, se descubrió como un modelo válido como optimizador. El desarrollo de la optimización con enjambres de partículas continúa, y en los últimos años, se han llevado a cabo numerosas investigaciones sobre la técnica y se ha aplicado con éxito en diferentes áreas, demostrando ser capaz de obtener mejores resultados de forma más rápida y con menor coste en comparación con otras estrategias.

De forma similar y como su propio nombre indica, la optimización mediante colonias de hormigas (*ant colony optimization*, ACO), encuentra su motivación en otro ejemplo de comunidad de la naturaleza: las colonias de hormigas reales y su comportamiento. El primer sistema de este tipo, llamado *Ant System*, fue desarrollado por Marco Dorigo en 1992 [199, 205], y desde 1995, se han propuesto varias ampliaciones del mismo, como el *Ant Colony System* del propio Dorigo junto con Gambardella [200, 201], o el *MAX-MIN Ant System* de Stützle y Hoos [202], aplicándose al problema del viajante con excelentes resultados. Además, Dorigo, Gambardella y Stützle han propuesto nuevas versiones híbridas con búsqueda local, y en 1999, Dorigo, Di Caro y Gambardella [203, 204] definieron la metaheurística de la optimización con colonias de hormigas.

Por otro lado, el término coevolución se refiere a los casos en los que dos (o más) especies influyen mutuamente en su evolución. Este concepto fue establecido por dos grandes biólogos evolucionistas de nuestro tiempo, Paul Ehrlich y Peter Raven, con su estudio sobre la evolución recíproca de mariposas y plantas, “*Butterflies and plants: A study in coevolution*”, publicado en 1965 [206]. Por su parte, Leigh Van Valen [207] lo desarrolló como la “Hipótesis de la Reina Roja”, en referencia a la frase de ésta en la obra de Lewis Carroll, “*Alicia en el país de las maravillas*”: “*It takes all the running you can do, to keep in the same place.*”

(Se necesita correr todo lo posible para mantenerse en el mismo lugar) Que en términos evolutivos, significa que la mejora continua de un sistema evolutivo es necesaria para sólo mantenerse al nivel de los demás sistemas con los que coevoluciona.

En la naturaleza, se presentan casos de coevolución tanto de antagonismo o rivalidad como de cooperación o mutualismo entre especies, y existen estrategias computacionales inspiradas en ambos tipos. Varios métodos de colaboración han sido diseñados para aplicarse en algoritmos coevolutivos-cooperativos [210-214] y aún se sigue trabajando en la búsqueda de métodos más eficaces. Sin embargo, es la coevolución competitiva la que más investigaciones ha motivado y ha sido de mayor trascendencia hasta la actualidad [215-221].

A continuación, se explican en profundidad algunos de los mecanismos principales de computación evolutiva, concretamente, los involucrados en este trabajo: los algoritmos genéticos, las estrategias evolutivas, la programación genética, la optimización con enjambres de partículas y la coevolución.

ALGORITMOS GENÉTICOS

Los algoritmos genéticos (*genetic algorithms*, *GA*), se centran en la resolución de problemas mediante la representación y manipulación de individuos de una población a nivel de genotipo. Dados un problema a resolver, un espacio de soluciones, un conjunto de operadores genéticos y un criterio de evaluación de las soluciones, el objetivo es encontrar el elemento del espacio de soluciones que sea óptimo según el criterio de evaluación. Es un proceso genérico de optimización, cuyas principales características distintivas son:

- La representación de soluciones mediante cadenas binarias o discretas análogas a las cadenas de ADN.
- El uso del cruce y la mutación para la obtención de variantes, de forma similar a la reproducción y mutación presentes en la naturaleza.
- La selección de posibles soluciones en proporción a sus resultados, inspirada en la selección natural de individuos según su adaptación al medio.

Un algoritmo genético consta de cuatro elementos fundamentales para su funcionamiento: cada individuo correspondiente al genotipo de una solución particular al problema, la población conformada por un conjunto finito de individuos, los operadores genéticos que transforman la población, y la función de evaluación, o de *fitness*, que mide la aptitud de las soluciones.

Dentro de los operadores, se incluye la selección, simulación computacional de la Selección Natural de Darwin, para la que existen distintos métodos como la ruleta, la selección jerárquica o los torneos. La ruleta selecciona a los individuos con una probabilidad directamente proporcional al *fitness* de los mismos, mientras que la selección jerárquica ordena los individuos en un ranking de mayor a menor *fitness* y son seleccionados con una probabilidad directamente proporcional a su puesto en el ranking. En la selección por torneos, con un tamaño *t* asignado, se escogen aleatoriamente *t* individuos y el de mayor *fitness* es seleccionado, proceso que se repite tantas veces como individuos tenga la población.

Pero como operadores genéticos estrictamente hablando, destacan el cruce y la mutación. El cruce simula la reproducción sexual en los organismos, y puede ser simple, multipunto o uniforme, dependiendo del tipo de combinación de los genes, pero en cualquier caso no se introducen nuevos alelos que no tuvieran los padres, perdiéndose variabilidad genética. Por tanto, para evitar la pérdida de alelos, se utiliza la mutación, que con una probabilidad determinada cambia un único gen de la cadena, pudiendo introducir cualquier alelo. De esta manera, la mutación evita la convergencia prematura, pero también aleatoriza la búsqueda, siendo conveniente una probabilidad de mutación alta al principio, para favorecer la exploración, y baja más adelante, cambiando a explotación a medida que se obtienen soluciones buenas. Se han desarrollado otros muchos operadores genéticos, como la inversión, pero no se utilizan con tanta frecuencia.

Otro factor a tener en cuenta es el tipo de inserción y reemplazo que se lleve a cabo entre generaciones. Según el tamaño de la población generada, las estrategias de inserción pueden dividirse en tres grupos. Se genera una población de igual tamaño que la anterior si la inserción es: generacional, por la que se renueva toda la población; generacional elitista, que conserva cierto número de los individuos más aptos y renueva el resto; o elitista pura, sólo se conserva el individuo más apto y se renueva el resto. La población generada es de menor tamaño con los mecanismos de: estado estacionario, que genera menos individuos; o estado estacionario puro, únicamente se genera un individuo. Por último, con una estrategia basada en el *fitness*, cierto número de los individuos más aptos se añaden a una población completamente renovada, generándose una población mayor que la anterior.

En cuanto a las estrategias de reemplazo, pueden ser globales o locales. En las globales se considera toda la población para escoger los reemplazados o bien de manera aleatoria, o los más antiguos, los peores o los más parecidos a los individuos a insertar. En las locales, por el contrario, sólo se consideran los individuos que se reproducen, siendo completamente reemplazados por los individuos nuevos, reemplazándose sólo el peor de los progenitores o los más parecidos a los generados.

En general, para eludir el estancamiento en mínimos locales, es necesario un ajuste adecuado de la presión selectiva, es decir, la tendencia del algoritmo a favorecer a los individuos más aptos, ya que, al reducir ésta, puede protegerse la diversidad genética pero a costa de la eficacia de cada iteración del algoritmo.

Ya expuestos los fundamentos de los algoritmos genéticos, su funcionamiento es el siguiente:

- 1 - Inicialización, generalmente aleatoria, de los individuos de la población.
- 2 - Evaluación de la población mediante la función de fitness.
- 3 - Selección de los individuos a reproducir.
- 4 - Reproducción de los individuos seleccionados: cruce, mutación...
- 5 - Inserción y reemplazo para obtener la nueva población.
- 6 - Si se cumple el criterio de parada, parar. Si no, volver al paso 2.

Como es lógico, un algoritmo genético no se ejecuta infinitamente, sino hasta que se cumple un criterio de parada definido. Dependiendo de si la detención del algoritmo está predeterminada (tras un tiempo o un número de iteraciones o de evaluaciones), o si puede variar entre ejecuciones diferentes (cuando no mejora, o la diversidad genética o la fenotípica caigan por debajo de un umbral), el criterio es de parada estática o dinámica, respectivamente.

ESTRATEGIAS EVOLUTIVAS

Las estrategias evolutivas (*evolutionary strategies, ES*), tienen una base similar a los algoritmos genéticos, pero la codificación de las cadenas o vectores, se realiza con valores continuos en lugar de discretos o binarios. Otras diferencias derivadas son:

- El uso de una mutación distinta con parámetros que evolucionan por sí mismos.

- Su potencia de búsqueda es la mutación, mientras que la del algoritmo genético es el cruce.

Además, existen varios tipos de estrategias evolutivas, atendiendo al tamaño de la población, hay de un solo individuo (1+1) o múltiples, que a su vez pueden ser de inclusión (μ, λ) o de inserción ($\mu + \lambda$), según su política de reemplazo.

En las estrategias evolutivas donde la población está compuesta por un único individuo, llamadas (1+1), este individuo varía de generación en generación y contiene dos vectores de números reales, uno de los cuales codifica una posible solución y otro que guarda las varianzas, involucradas en la mutación.

La mutación permite la creación de un nuevo individuo, produciéndose en cada generación, una población intermedia de dos individuos. En cada parte del individuo, el operador de mutación actúa de una manera: en la funcional se aplica una mutación Gaussiana, mientras que el vector de varianzas se muta mediante la regla del 1/5.

Para realizar la mutación del vector de codificación, se le añade un valor aleatorio de una distribución Gaussiana. Este valor estará más próximo al cero cuanto menor sea la varianza σ , es decir, cuando más cerca se esté de la solución óptima. La fórmula de esta operación, es la siguiente, donde x hace referencia a cada valor del vector de codificación:

$$x_2 = x_1 + N_o(\sigma_1)$$

La modificación de la varianza acorde con la cercanía de la solución actual respecto de la solución óptima se debe a la mutación del vector de varianzas. Ésta debe realizarse de manera que, cuando el individuo esté lejos de la solución óptima, las varianzas adquieran valores grandes, en cambio, de encontrarse cerca de la solución, deben obtenerse valores pequeños de varianza. Para este efecto, se emplea la regla del 1/5, por la cual es necesario guardar la proporción de las veces que el individuo, y en este caso la población, ha mejorado en las últimas ejecuciones de la mutación. A partir de esta cantidad, se aplica que:

- Si es menor que $1/5$, la varianza se incrementa, multiplicándose por un parámetro c_i mayor que 1, habitualmente, igual a 1,18.

- Si es mayor que $1/5$, la varianza debe decrementarse, multiplicándose entonces por un parámetro c_d menor que 1, que normalmente toma el valor de 0,82.

- Si, por el contrario, la proporción de mejoras es igual a $1/5$, se mantiene la varianza tal cual.

En cuanto a la proporción de veces que la población ha mejorado, bastaría con tomar la proporción de veces que la población ha cambiado, pues es equivalente. Esto es debido a que la política de reemplazo, en este tipo de estrategias evolutivas, consiste en quedarse con el mejor individuo de la población intermedia, es decir, el mejor individuo entre el inicial y el generado.

Por consiguiente, el procedimiento general que siguen las estrategias evolutivas de un solo individuo puede resumirse en:

- 1 - Inicialización aleatoria del vector de valores continuos y el vector de sus varianzas, este último preferiblemente con valores grandes.
- 2 - Evaluación del individuo según la función de fitness.
- 3 - Mutación del individuo para generar un nuevo individuo.
- 4 - Evaluación del nuevo individuo.
- 5 - Proseguir con el individuo de mayor fitness y actualizar la proporción de mejoras si es necesario.
- 6 - Mutación del vector de varianzas siguiendo la regla del $1/5$.
- 7 - Si se cumple el criterio de parada, terminar. Si no, continuar desde el paso 3.

Pudiendo utilizarse como criterios de parada cualquiera de los indicados para los algoritmos genéticos.

No obstante, si el problema a abordar requiere poblaciones con más de un individuo, entonces habría que aplicar las estrategias evolutivas múltiples, para las que el procedimiento cambia. Por un lado, el almacenamiento de una estadística de cambios para cada individuo se vuelve muy costoso, por lo que se prescinde de la mutación mediante la regla del $1/5$. Pero por otro lado, se posibilita la utilización del operador de cruce, precisándose de una nueva definición de ambas operaciones.

En referencia al cruce, también se diferencia, aunque ligeramente, su función sobre la parte correspondiente a la codificación y el vector de

varianzas de cada individuo. Mientras que un nuevo vector de codificación se obtiene a partir de la media de los vectores de de sus progenitores, las nuevas varianzas se generan con una cuenta similar que sustituye la división propia de la media por una raíz cuadrada.

Todos los cálculos se realizan emparejando los valores de posiciones análogas, combinando los genes para producir alelos nuevos. Es por eso que este tipo de cruce recibe el nombre de cruce uniforme.

En el caso de la mutación, se conserva la manera de actuar sobre la parte funcional, pero la actualización del vector de varianzas ya no se basa en la regla del 1/5, y pasa a disminuir siempre, de acuerdo a la fórmula:

$$\sigma' = e^{N(o,r) \cdot \sigma}$$

Además, al tratarse de estrategias evolutivas múltiples, se puede utilizar el operador de selección, concretamente, por muestreo aleatorio simple con probabilidad proporcional a la aptitud de los individuos, es decir, a través del método de la ruleta o alguno similar.

Igualmente, el aumento de tamaño de la población afecta a las políticas de inserción y reemplazo, aumentándose las alternativas. En este aspecto, las estrategias evolutivas pueden seguir dos políticas distintas: el reemplazo por inclusión y el reemplazo por inserción. En ambas, la población resultante tiene tantos individuos como número de progenitores (μ).

En el reemplazo por inclusión ($\mu+\lambda$), la población resultante se completa con los μ mejores individuos tanto progenitores como descendientes. Así, los individuos progenitores y descendientes compiten entre sí para pasar a la siguiente generación.

Por su parte, dentro del reemplazo por inserción (μ,λ), con el que los progenitores menos aptos siempre son sustituidos por los descendientes, existen a su vez dos posibilidades: si hay tantos descendientes (λ) como individuos a completar en la población resultante (μ), se incorporan todos los descendientes; pero si el total de los descendientes no completa la siguiente población, entonces se insertan también los $\mu-\lambda$ mejores progenitores.

A continuación, se describe el funcionamiento de las estrategias evolutivas múltiples, indicándose el paso que difiere dependiendo de la política de reemplazo empleada.

- 1 - Inicialización de la población de individuos y sus vectores de valores continuos y de varianzas, éstas con valores grandes.
- 2 - Evaluación de la población según la función de fitness.
- 3 - Selección con muestreo aleatorio proporcional de los individuos a reproducir.
- 4 - Reproducción de los individuos seleccionados mediante cruce.
- 5 - Mutación de los λ individuos generados.
- 6 - $(\mu+\lambda)$ - Obtención de una nueva población mediante la inclusión de los μ mejores individuos de la población intermedia formada por progenitores y descendientes.
- 7 - (μ,λ) - Obtención de una nueva población mediante la inclusión de los λ descendientes, y los progenitores a excepción de los λ peores.
- 8 - Si se cumple algún criterio de parada, terminar. Si no, seguir desde el paso 2.

De nuevo, coincidiendo con las estrategias evolutivas de un solo individuo, se pueden utilizar como criterios de parada cualquiera de los indicados previamente para los algoritmos genéticos.

PROGRAMACIÓN GENÉTICA

La programación genética (*genetic programming, GP*), es una técnica cada vez más popular que comparte fundamentos con el algoritmo genético, pero a diferencia de éste, la codificación de las soluciones no toma la forma de cadenas, sino de árboles con sintaxis de Lisp. Los árboles son de tamaño variable y cada uno entero representa una función que puede ser evaluada como posible solución a un problema. Los nodos del árbol contienen una función u operador, y las hojas, un terminal u operando, habiendo un conjunto limitado de símbolos disponibles tanto de funciones como de terminales.

Para inicializar una población con individuos de este tipo, es necesario asignar la probabilidad con la que un nodo contendrá una función o un terminal, respectivamente. Si se decide, mediante dicha probabilidad, que un nodo sea de función, se coloca una de las disponibles y se mantiene como nodo. Si, por el contrario, el nodo debe corresponder a un terminal, entonces se escoge uno de los terminales y constantes disponibles y el nodo pasa a ser hoja.

Con esta clase de representación de soluciones, la evaluación de las mismas supone otorgar unos valores, acordes con un ejemplo del problema

a resolver, a los símbolos terminales y comprobar cuánto se separa cada solución obtenida de la solución esperada para ese ejemplo concreto, tomándose esta distancia como el fitness de cada individuo.

Una vez generada la población de árboles, los operadores genéticos que se pueden aplicar son el cruce y la mutación. Por un lado, el cruce consiste en el intercambio de un subárbol cualquiera de un progenitor con otro subárbol aleatorio de otro progenitor. Así, el cruce en programación genética presenta la capacidad de generar nuevos individuos mediante individuos idénticos.

Por otro lado, la mutación consiste en producir un árbol nuevo a partir de un único árbol progenitor, y presenta tres variaciones posibles: la mutación terminal simple, la mutación funcional simple y la mutación de árbol. Para aplicar la mutación terminal simple, se selecciona aleatoriamente un terminal del progenitor y se sustituye por otro distinto de entre los disponibles. De igual modo, en la mutación funcional simple, una función aleatoria del árbol progenitor es reemplazada por otra del conjunto de funciones permitidas. Finalmente, la mutación de árbol consiste en la sustitución de un subárbol cualquiera del individuo por otro subárbol nuevo generado aleatoriamente.

En conclusión, el procedimiento a seguir por un algoritmo genérico de programación genética quedaría de la siguiente manera:

- 1 - Inicialización aleatoria de la población de árboles.
- 2 - Evaluación de cada árbol.
- 3 - Insertar los mejores individuos en una población intermedia.
- 4 - Reproducir un individuo o dos de la población inicial, mediante mutación o cruce, respectivamente.
- 5 - Incluir el/los nuevos individuos en la población intermedia.
- 6 - Si la población intermedia está incompleta, volver al paso 4. Si no, reemplazar la población inicial con la población intermedia.
- 7 - Si se cumple algún criterio de parada, terminar. Si no, repetir desde el paso 2.

Al igual que en las otras técnicas ya descritas, los criterios de parada apropiados para algoritmos genéticos, también son válidos para programación genética.

A pesar de que la búsqueda de relaciones matemáticas puede resultar su aplicación más intuitiva, actualmente, la programación genética no está limitada a la misma, habiendo trabajos de investigación relativos a otras áreas como la cinemática inversa (*inverse kinematics*), la navegación, o el diseño automático de estructuras complejas.

OPTIMIZACIÓN CON ENJAMBRES DE PARTÍCULAS

La optimización con enjambres de partículas (*particle swarm optimization, PSO*), es una técnica computacional de inspiración biológica, donde los individuos de una población tratan de moverse hacia la mejor posición conocida por ellos y por los demás individuos de la población o de su mismo círculo social, siguiendo el objetivo de maximizar o minimizar una función de evaluación.

Este modelo asume el intercambio de información, a partir de interacciones sociales, entre los individuos o agentes de búsqueda. El comportamiento de éstos se rige por tres reglas principales: separación, alineamiento y cohesión. Es decir, cada agente trata de mantenerse suficientemente separado de sus vecinos, a la par que orientarse según su orientación media y desplazarse hacia la posición media de los mismos.

La codificación de cada solución propuesta, es un vector con las coordenadas de cada agente o partícula, que se va modificando con la velocidad de la partícula. Todas las partículas llevan una velocidad distinta de cero, nunca dejan de “volar”, y la propia velocidad también se actualiza. Además de su posición actual y su velocidad, cada partícula memoriza y comparte la mejor solución encontrada por sí misma, y conoce la mejor solución encontrada por toda la población o enjambre.

De aquí se extrae el principio fundamental de la dinámica de los enjambres de partículas: el balance entre la influencia de la mejor solución global y la mejor solución individual encontradas. El movimiento de cada partícula, es decir, cómo cambien sus coordenadas, se determina con su vector de velocidad, según la siguiente fórmula, donde x es el vector de posición y v el vector de velocidad:

$$x(t + 1) = x(t) + v(t)$$

Es en la actualización del vector de la velocidad donde se tienen en cuenta las coordenadas de las mejores soluciones obtenidas, tanto la global como la individual:

$$v(t + 1) = \overbrace{w \cdot v(t)}^{\text{Inercia}} + \overbrace{\phi_1 \cdot r(o, a_1) \cdot (x^{max}(t) - x(t))}^{\text{Memoria individual}} + \overbrace{\phi_2 \cdot r(o, a_2) \cdot (G(t) - x(t))}^{\text{Memoria colectiva}}$$

Donde $r(o, a)$ es un número aleatorio entre 0 y a ; x^{max} es la posición de la mejor solución individual de la partícula y G , de la global; y w , ϕ_1 y ϕ_2 son pesos. Concretamente, w es el peso de la inercia, y puede ir disminuyéndose en cada iteración para pasar de exploración a explotación, puesto que valores altos de w favorecen la búsqueda global, mientras que valores bajos favorecen la búsqueda local. También, pueden asociarse pesos a las otras dos componentes, correspondientes a la memoria individual y a la memoria global, para reflejar la importancia de cada una. La combinación de todos estos parámetros determina las propiedades de convergencia del algoritmo.

Además, para mantener la concordancia con la realidad del problema que se esté tratando de resolver, se define una velocidad máxima v_{max} , y en aplicaciones prácticas, también se establecen los límites de la posición de las partículas, de acuerdo a la naturaleza del espacio de búsqueda y sus límites.

Hasta aquí se ha hablado de la memoria global de toda la población, pero se puede introducir el concepto de vecindario de una partícula, formado por las partículas vecinas que se pueden definir conforme a diferentes topologías. En caso de llevar a cabo el algoritmo considerando los vecindarios de cada partícula en vez de la población en general, en el componente de la memoria colectiva se sustituiría G por L , la mejor solución encontrada por el vecindario, pasando a ser la memoria social local en el cálculo de la velocidad.

Independientemente del uso o no de vecindario, el funcionamiento de la optimización con enjambres de partículas es el mismo, y consta de los siguientes pasos:

- 1 - Inicialización de la población de partículas distribuidas aleatoriamente.
- 2 - Evaluación de cada partícula con la función objetivo.
- 3 - Actualización de memoria individual y colectiva, si se han mejorado respectivamente.
- 4 - Modificación del vector de velocidad de cada partícula.
- 5 - Desplazamiento de cada partícula según su velocidad.
- 6 - Si se cumple algún criterio de parada, terminar. Si no, continuar desde el paso 2.

Por último, mencionar que los cálculos descritos previamente corresponden al PSO básico de mayor repercusión, el que utiliza valores continuos, pero hay otra versión desarrollada con valores binarios [223-228], que no ha captado tanta atención en investigaciones anteriores.

COEVOLUCIÓN

La coevolución (*co-evolution*, *CE*), es una técnica con una fuerza importante cuando se trata de generar adaptaciones muy complejas. Como se ha mencionado previamente, existen dos tipos de coevolución, computacionalmente hablando: la coevolución competitiva, donde dos sistemas independientes compiten por recursos compartidos y el éxito o supervivencia de uno supone el fracaso o desaparición del otro, como sucede en la relación entre depredador y presa; y la coevolución cooperativa, en la que dos sistemas independientes cooperan para aumentar sus posibilidades de supervivencia, como la simbiosis.

La categoría más extendida es la coevolución competitiva, en la que se genera una población de ejemplos, además de la de soluciones, y a semejanza de lo que ocurre con un depredador y su presa, si los individuos de una población evolucionan, los individuos de la otra población reciben una presión mayor para evolucionar más. Así, el mecanismo se realimenta y en ambas poblaciones se producen individuos cada vez mejores.

Este sistema surge para solucionar el problema del entrenador ideal, es decir, la elección del conjunto de entrenamiento idóneo en los algoritmos evolutivos. Lo habitual en los problemas objetivo, es que se disponga de una cantidad inmensa de posibles ejemplos de entrenamiento, pero para que el proceso evolutivo sea eficiente sólo puede utilizarse una muy pequeña proporción. Aquí aparece el conflicto de decidir qué ejemplos conformarían el mejor conjunto de evaluación, controlando el riesgo de que se produzca especialización u *overfitting*, y teniendo en cuenta que un conjunto puede dar buenos resultados para unos individuos y malos para otros.

La solución que plantea la coevolución es la generación de dos poblaciones que se evalúen mutuamente: al evaluar un individuo solución con un individuo conjunto de entrenamiento, el fitness del individuo conjunto será inversamente proporcional al fitness obtenido por el individuo solución. Así, una evaluación buena para el individuo solución, corresponderá a una evaluación mala para el individuo conjunto y viceversa. Puesto que cada individuo se evalúa con varios distintos de la

otra población, su evaluación total se constituye con la suma u otra medida estadística que englobe todas sus evaluaciones individuales.

Continuando en el contexto de un algoritmo evolutivo, los operadores genéticos se aplican con normalidad sobre los individuos de la población de soluciones, mientras que sobre los individuos pertenecientes a la población de conjuntos de entrenamiento sólo se efectúa la selección, dándose prioridad a los individuos que peor hayan sido resueltos por la población de soluciones. De esta manera, los conjuntos más difíciles tienen mayor probabilidad de ser seleccionados para la evaluación de las soluciones, y al utilizarse una ventana temporal para la misma, las dos poblaciones se adaptan la una a la otra.

Una vez incorporadas estas características al procedimiento de un algoritmo genético, el ciclo coevolutivo se compone de los siguientes pasos:

- 1 - Inicialización aleatoria de la población de individuos correspondientes a posibles soluciones y de la población de individuos con conjuntos de ejemplos.
- 2 - Evaluación de la población de soluciones.
- 3 - Definición del número de encuentros a realizar, n , con un valor pequeño respecto al tamaño de las poblaciones. Y durante n veces:
 - 3.1 - Selección de un individuo solución s y un individuo conjunto c .
 - 3.2 - Encuentro de evaluación entre s y c , y actualización de sus respectivos fitness.
- 4 - Selección de dos individuos solución a reproducir.
- 5 - Reproducción de los dos individuos mediante cruce y mutación, para obtener un descendiente d .
- 6 - Definición del número de encuentros a realizar, m , con un valor pequeño respecto al tamaño de las poblaciones. Y durante m veces:
 - 6.1 - Selección de un individuo conjunto c .
 - 6.2 - Encuentro de evaluación entre d y c , y actualización de sus respectivos fitness.
- 7 - Si el descendiente d es más apto que el peor de la población de soluciones, el peor es reemplazado por d .
- 8 - Si se cumple algún criterio de parada, terminar. Si no, repetir desde el paso 3.

Los algoritmos coevolutivos de este tipo incluyen diversas aplicaciones, como tareas de clasificación, control de procesos, planificación de rutas o satisfacción de restricciones.

Aparte de la coevolución competitiva, existe otra variante del mecanismo de estos algoritmos que se aleja del modelo depredador-presa y se asemeja al fenómeno de la simbiosis, produciéndose una retroalimentación del fitness positiva entre las dos poblaciones. Este grupo recibe el nombre de algoritmos coevolutivos cooperativos, y fueron propuestos originalmente por Potter y de Jong en 1994 [210]. En este caso, ambas poblaciones están integradas por individuos, llamados especies, que forman soluciones al combinarse, y el fitness de cada individuo que se ha combinado depende de la aptitud de la solución generada. Un planteamiento de este modelo es el *Symbiotic Coevolutionary Algorithm* (SCA), ideado en 2005 por Wallin et al. [212], que establece una analogía entre la relación simbiótica de un huésped con su parásito y las dos poblaciones de especies. De esta forma, una población está compuesta por individuos huéspedes que equivalen a soluciones completas, y la otra, por individuos parásitos o soluciones parciales. Mediante la asociación de cada huésped con un parásito, se conforman los candidatos a reproducirse y generar nuevos huéspedes. Los parásitos no se reproducen por sí solos, únicamente mutan.

Como se ha podido ver, la computación evolutiva es un área que continúa considerablemente abierta a la investigación, y entre las líneas de estudio que se espera seguir en un futuro están: la auto-adaptación de los algoritmos, evitando el uso de parámetros ad-hoc; la explotación de arquitecturas paralelas, para favorecer el ahorro de esfuerzo computacional; un mayor conocimiento teórico, de los modelos matemáticos de los algoritmos y sobre las propiedades de convergencia, epístasis y diversidad genética; una mejor comprensión de la evolución natural, con simulaciones computacionales de las interacciones entre seres vivos más complejas; o el algoritmo genético que pueda prescindir de parámetros.

Para concluir, entre los múltiples campos de aplicación de la computación evolutiva y de inspiración biológica, se encuentran: la paralelización automática de programas secuenciales, el diseño tecnológico y de instalaciones, la optimización de procesos de producción y de redes de comunicaciones, la predicción de mercados, el diagnóstico médico automático, los sistemas automáticos de decisión, el análisis de datos, etc.

2.3 Análisis

Una vez planteado el problema a tratar, se ha realizado el análisis de todos los aspectos del mismo, extrayéndose, a partir de los objetivos planteados, una serie de requisitos que se exponen a continuación. Estos requisitos se refieren a los módulos a implementar para el desarrollo de las fases de aprendizaje y clasificación, y, según el estándar de la ESA, corresponden a los requisitos de capacidad, englobados dentro de los requisitos de usuario.

De cada requisito se describen las siguientes características, recogidas en una tabla genérica que también se indica a continuación:

Tabla 2: Genérica de requisitos

Identificador	<Grupo>-<Subgrupo>-<Número>
Nombre	
Descripción	
Prioridad	

- El **identificador** de un requisito es una clave única que se compone de: el número del requisito respecto al total; el grupo principal, es decir, requisito de usuario; y el subgrupo, que, en este caso, corresponderá a requisitos de capacidad para todos.
- El **nombre**, enuncia breve y significativamente en qué consiste el requisito.
- La **descripción** del requisito, explicada más extensamente.
- La **prioridad** de un requisito puede ser alta, media o baja, representando con qué urgencia debe integrarse, según la relevancia del mismo dentro del sistema a implementar.

2.3.1 Requisitos de usuario

Los requisitos de usuario expresan qué debe hacer un sistema desde una perspectiva a alto nivel.

REQUISITOS DE CAPACIDAD

Dentro de los requisitos de usuario, los requisitos de capacidad engloban las facultades generales que tendrá el sistema, básicamente, hacen referencia a qué podrá hacer el sistema en un contexto práctico.

Tabla 3: Requisitos de capacidad

Identificador	RU-RC-01
Nombre	Preprocesamiento de datos
Descripción	Preprocesamiento de las señales de los sensores, tratando valores desconocidos, ruido, etc.
Prioridad	Alta

Identificador	RU-RC-02
Nombre	Generación de conjuntos
Descripción	Generación de conjuntos de test y entrenamiento para cada tipo de problema definido
Prioridad	Alta

Identificador	RU-RC-03
Nombre	Clustering óptimo de instancias
Descripción	Encontrar la agrupación óptima de datos de actividades según similitud de instancias, en clusters, mediante un enjambre de partículas
Prioridad	Alta

Identificador	RU-RC-04
Nombre	Representación clusters
Descripción	Representar internamente la configuración de los clusters, lo más simplemente posible
Prioridad	Media

Identificador	RU-RC-05
Nombre	Exportación de configuración de clusters
Descripción	Exportar la configuración de los clusters, con una codificación simple
Prioridad	Media

Identificador	RU-RC-06
Nombre	Distribución de clusters
Descripción	Distribuir instancias en los clusters definidos por la configuración final
Prioridad	Alta

Identificador	RU-RC-07
Nombre	Exportación de clusters
Descripción	Exportar la distribución de instancias en los clusters definidos por la configuración final
Prioridad	Media

Identificador	RU-RC-08
Nombre	Importación de cluster
Descripción	Importar las instancias pertenecientes a un cluster de los producidos.
Prioridad	Media

Identificador	RU-RC-09
Nombre	Aprendizaje
Descripción	Aprendizaje de patrones de comportamiento de datos de actividades, mediante un algoritmo evolutivo
Prioridad	Alta

Identificador	RU-RC-10
Nombre	Representación clasificadores
Descripción	Representación interna de modelos de clasificación, o sea, los clasificadores-solución, mediante árboles de decisión.
Prioridad	Alta

Identificador	RU-RC-11
Nombre	Exportación de clasificadores
Descripción	Exportación de los clasificadores, es decir, árboles de decisión, con una codificación lo más simple posible.
Prioridad	Media

Identificador	RU-RC-12
Nombre	Importación de configuración de clusters
Descripción	Importar la configuración de los clusters
Prioridad	Alta

Identificador	RU-RC-13
Nombre	Importación de clasificadores
Descripción	Importación de los clasificadores, es decir, árboles de decisión, transformándolos de la codificación simple a la representación interna.
Prioridad	Alta

Identificador	RU-RC-14
Nombre	Importación de conjunto de test
Descripción	Lectura y procesamiento de las instancias contenidas en el conjunto de test.
Prioridad	Alta

Identificador	RU-RC-15
Nombre	Distribución de instancias de test
Descripción	Distribución de las instancias, pertenecientes al conjunto de test, entre los clusters importados.
Prioridad	Alta

Identificador	RU-RC-16
Nombre	Predicción de actividades
Descripción	Predicción de actividades para las instancias de cada cluster, con el clasificador vinculado a cada uno.
Prioridad	Alta

Identificador	RU-RC-17
Nombre	Evaluación
Descripción	Evaluación de los resultados obtenidos comparando las predicciones generadas con las etiquetas reales de las instancias.
Prioridad	Alta

3. Diseño e implementación

En este capítulo, se detalla el diseño del sistema y las decisiones tomadas al respecto, con ayuda de diagramas de flujo y diagramas de clases, y, seguidamente, se explica en profundidad los componentes implementados. Para empezar, se ha decidido que el sistema esté compuesto por tres módulos:

- Un **algoritmo evolutivo (AE)** que entrene y genere los clasificadores de actividades. Concretamente, la representación escogida para los clasificadores son los árboles de decisión.
- Un **enjambre de partículas (PSO)** que agrupe las instancias de aprendizaje en clusters según su afinidad, mediante la optimización de la configuración de los centroides.
- Un **programa evaluador (PEv)** que reúna los productos de todo lo anterior para la simulación de la clasificación final del sistema de cara a nuevos datos de realización de actividades.

Así, una vez implementado el sistema, el mecanismo se puede observar en la *Ilustración 2* y consiste en: (1) la ejecución del módulo de clustering, es decir, el enjambre de partículas del que se obtienen los distintos clusters de ejemplos y sus respectivos centroides; (2) después, la fase de aprendizaje con el algoritmo evolutivo, creándose un árbol clasificador para cada uno de los clusters de ejemplos; y (3) la introducción de los valores de los centroides y los árboles de cada cluster en el programa evaluador para aplicar, a cada instancia nueva o de test, el árbol de decisión correspondiente al cluster al que pertenezca la misma.

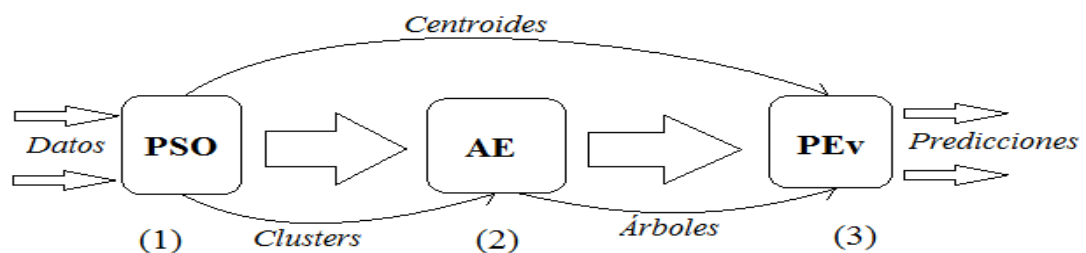


Ilustración 2: Mecanismo del sistema

3.1 Funcionamiento general de los módulos

En el caso del módulo 1, se trata de la agrupación de los datos de entrada mediante la minimización, con un enjambre de partículas, de la distancia de cada ejemplo al centroide del cluster al que pertenece. La pertenencia a un cluster por parte de una instancia se decide por la distancia de los atributos de ésta a los centroides de los clusters, escogiéndose el cluster cuyos centroides resulten estar más cercanos a la instancia. A la optimización con un enjambre de partículas se le añade una modificación, agregándose un factor más al movimiento de las partículas: la media de las instancias de cada cluster. De esta manera, se emplea una combinación del algoritmo genérico de los enjambres de partículas y el algoritmo de clustering k-medias.

Por su parte, el módulo 2 consiste en el entrenamiento de árboles de decisión para clasificar con éxito las actividades de los datos. Para este fin, se aplica un algoritmo evolutivo conformado por una fusión entre la programación genética, perfecta para la evolución de los árboles de decisión, y las estrategias evolutivas, muy útiles para la adaptación de las comparaciones de los nodos, dado que los atributos son de tipo numérico. En este caso, se toman como entradas los diferentes grupos de instancias generados en el módulo 1, y a cada uno se le aplica el algoritmo evolutivo durante un cierto número de iteraciones, evolucionando una población de árboles-solución mediante los operadores genéticos propios de la programación genética y las estrategias evolutivas. De aquí, se obtienen los árboles de decisión ligados a cada cluster.

Además, para abordar el asunto de la elección de un conjunto de entrenamiento adecuado, en ambos módulos se integra la técnica de la coevolución, incorporándose una población de conjuntos de ejemplos que va evolucionando a la par que haciendo evolucionar a la otra población o al enjambre de partículas. Se ha seleccionado un mecanismo distinto de coevolución para cada módulo, los cuales se explican más adelante en este capítulo.

Por otro lado, el módulo 3, es decir, el programa destinado a evaluar el modelo final generado, recibe los centroides producidos en la fase de clustering (1) y los árboles de decisión generados en la fase de aprendizaje (2). A partir de los centroides, el programa reparte los ejemplos desconocidos o de test, entre los diferentes clusters, y, una vez analizados y traducidos los árboles de decisión, éstos son aplicados a los ejemplos del cluster al que están vinculados respectivamente. Ya realizadas las predicciones oportunas, el programa evalúa los resultados obtenidos mediante unas métricas a definir.

3.2 Diagramas de flujo

Para aclarar el funcionamiento seguido por cada uno de los módulos, éste se presenta de manera visual, en un diagrama de flujo a alto nivel junto a una lista descriptiva de los pasos que lo forman, para cada módulo.

Módulo 1: PSO

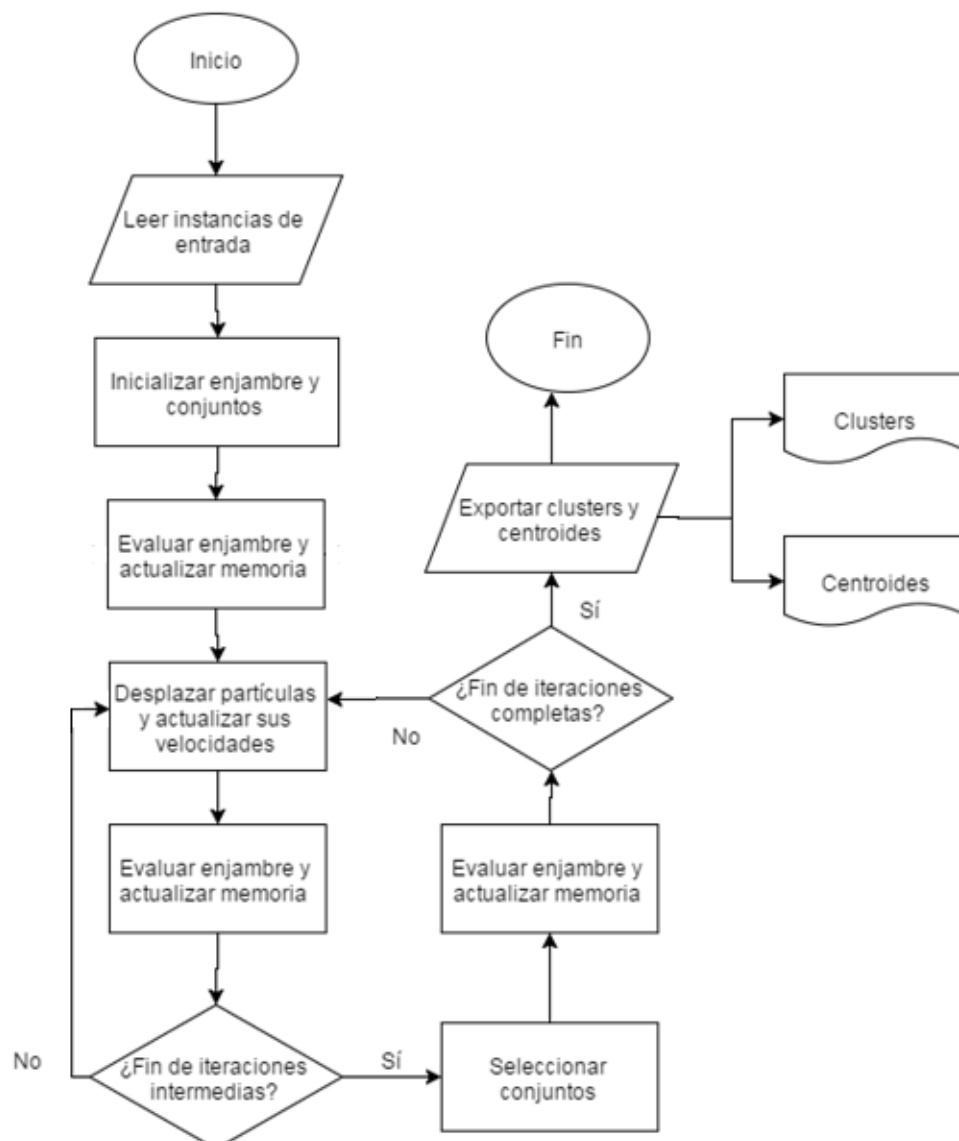


Ilustración 3: Diagrama de flujo Módulo 1: PSO

- 1 - Obtención de las instancias a partir de la lectura del fichero de datos de entrada.
- 2 - Distribución de todas las instancias en conjuntos fijos disjuntos, que conforman una población de conjuntos de entrenamiento.
- 3 - Inicialización de una ventana temporal con un número reducido de conjuntos de la población.
- 4 - Inicialización dirigida de las partículas correspondientes a configuraciones distintas de los centroides de los clusters.
- 5 - Evaluación del enjambre con respecto a los conjuntos de la ventana, y viceversa, actualizando la memoria de la mejor posición de cada partícula y del enjambre en global, si fuera necesario.
- 6 - Desplazamiento de las partículas según su velocidad, que se actualiza de acuerdo a los factores de la inercia y la memoria individual de cada partícula, la memoria global del enjambre y la media de cada cluster representado en la partícula.
- 7 - Evaluación del enjambre con respecto a los conjuntos de la ventana, y viceversa, actualizando la memoria de la mejor posición de cada partícula y del enjambre en global, si fuera necesario.
- 8 - Actualización de los pesos de inercia y media para iteraciones intermedias.
- 9 - Si se han cumplido las iteraciones intermedias, continuar al siguiente paso. Si no, ejecutar desde el paso 6.
- 10 - Selección de los conjuntos de la población para insertar en una nueva ventana. Se seleccionan los “mejores” conjuntos, es decir, con los que el enjambre haya obtenido peores resultados. Se reemplazan todos los conjuntos de la ventana anterior, aunque pueden haber sido seleccionados otra vez por estar entre los mejores.
- 11 - Evaluación del enjambre con respecto a los conjuntos de la ventana, y viceversa, actualizando la memoria de la mejor posición de cada partícula y del enjambre en global, si fuera necesario.
- 12 - Si se han realizado todas las iteraciones completas, continuar al siguiente paso. Si no, ejecutar desde el paso 6.
- 13 - Realizar el reparto de todas las instancias en los clusters finales, o sea, los de la mejor partícula del enjambre.

- 14 -Imprimir en un fichero la configuración de centroides de la mejor partícula final.
- 15 -Por cada cluster, exportar las instancias pertenecientes al mismo en un fichero.

Módulo 2: AE (para cada cluster de entrada)

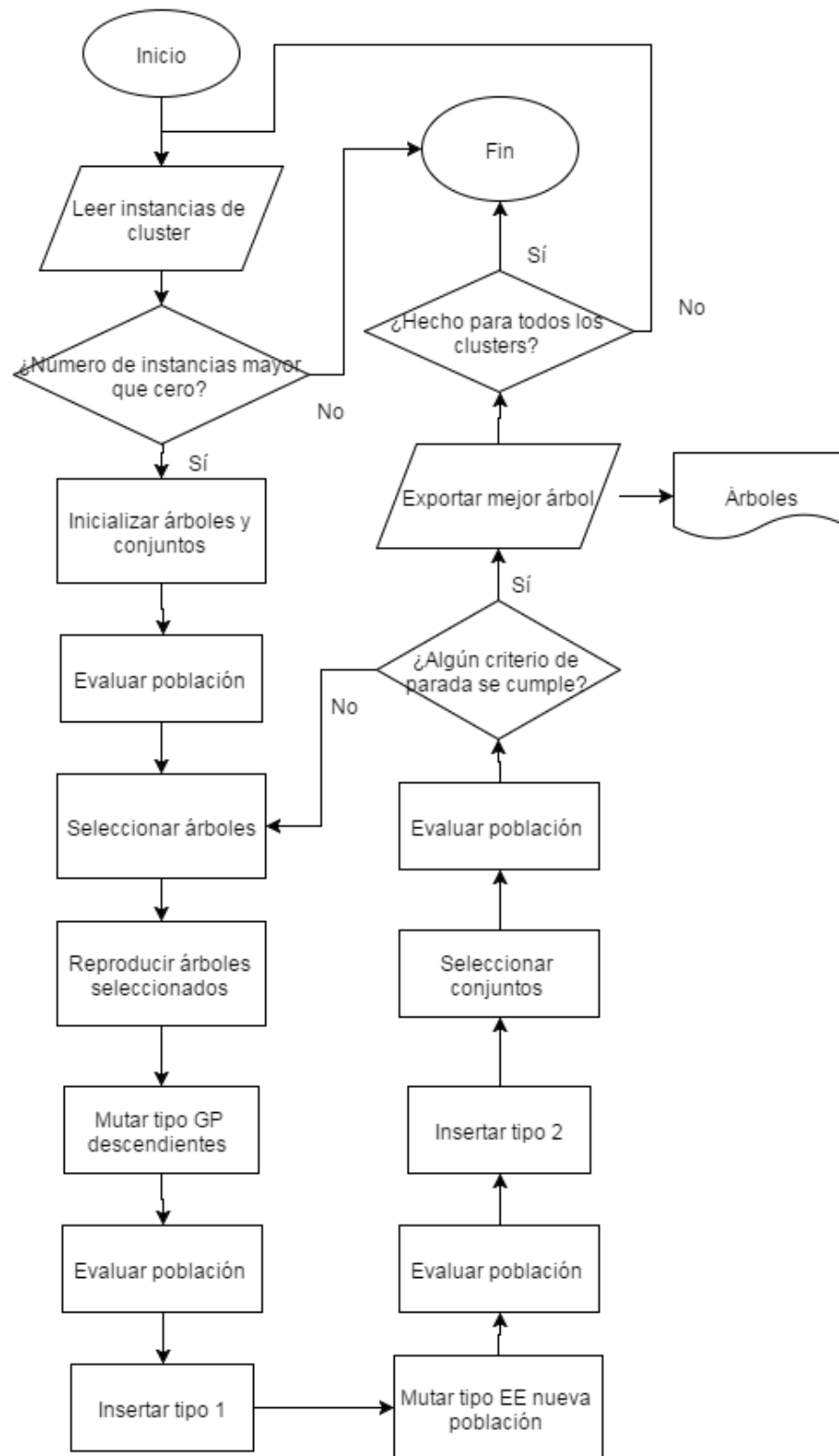


Ilustración 4: Diagrama de flujo Módulo 2: AE

- 1 - Obtención de las instancias a partir de la lectura del fichero de datos del cluster correspondiente.
- 2 - Si el número de instancias del cluster es mayor que cero, continuar al siguiente paso. Si no, terminar la ejecución para este cluster.
- 3 - Cálculo de la probabilidad con la que aparecen las distintas clases en las instancias del cluster.
- 4 - Inicialización de un número reducido de conjuntos de entrenamiento que se incluyen en una ventana temporal.
- 5 - Inicialización dirigida de los individuos con forma de árboles de decisión que, a su vez, conforman la población de soluciones del problema de clasificación. → Población inicial.
- 6 - Evaluación de la población de árboles inicial con respecto a los conjuntos de la ventana, y viceversa.
- 7 - Selección de mejores árboles a reproducir.
- 8 - Reproducción de los árboles seleccionados.
- 9 - Mutación de árbol, propia de la programación genética (GP), de los descendientes generados.
- 10 - Mutación simple, propia de la programación genética (GP), de los descendientes generados. → Población intermedia 1.
- 11 - Evaluación de la población de árboles intermedia 1 con respecto a los conjuntos de la ventana, y viceversa.
- 12 - Inserción, tipo 1, del mejor individuo de la población inicial en sustitución del peor de la población intermedia 1, en caso de ser mejor el primero. → Población final 1.
- 13 - Mutación de los valores continuos contenidos en los nodos de los árboles de la población final 1 mediante el uso de varianzas, operador propio de las estrategias evolutivas. → Población intermedia 2.
- 14 - Evaluación de la población de árboles intermedia 2 con respecto a los conjuntos de la ventana, y viceversa.
- 15 - Inserción del mejor individuo de la población final 1 en sustitución del peor de la población intermedia 2, si el primer individuo es mejor que el mejor individuo de la población intermedia 2. → Población final 2.

- 16 -Selección de un porcentaje determinado de conjuntos de ejemplos de la ventana actual para insertarlos en la nueva ventana. Se seleccionan los “mejores” conjuntos, es decir, con los que la población de soluciones haya obtenido peores resultados. Los huecos restantes se reemplazan con nuevos conjuntos recién inicializados.
- 17 -Evaluación de la población de árboles intermedia 2 con respecto a los conjuntos de la ventana, y viceversa.
- 18 -Si se cumple algún criterio de parada, continuar al siguiente paso. Si no, ejecutar desde el paso 7 tomando la población final 2 como población inicial.
- 19 -Exportar el mejor árbol de la población final 2 en versión simplificada, en un fichero.

Módulo 3: PEv

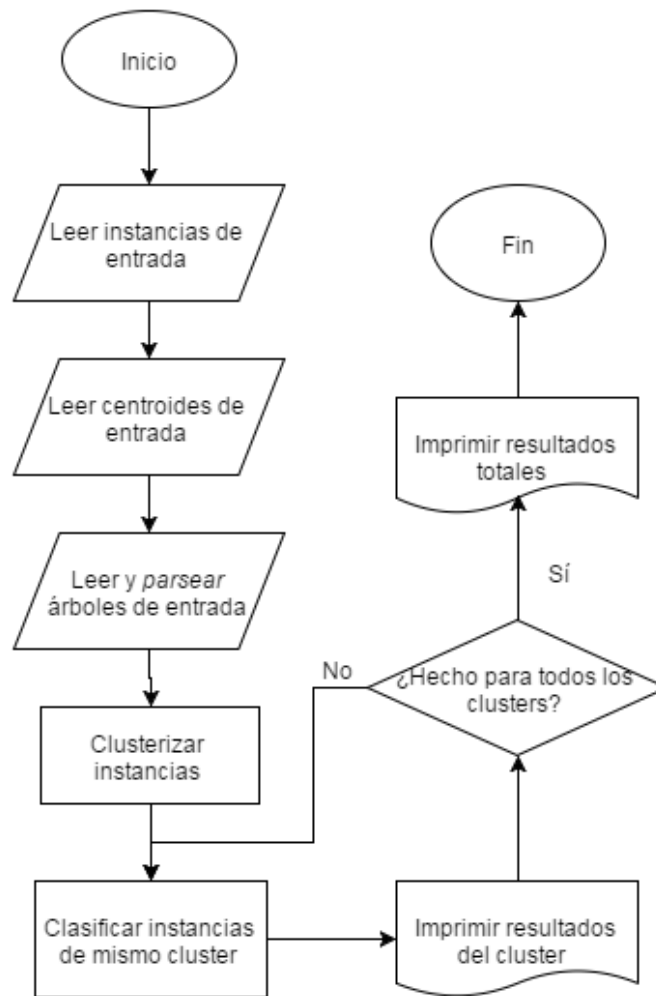


Ilustración 5: Diagrama de flujo Módulo 3: PEv

- 1 - Obtención de las instancias desconocidas a partir de la lectura del fichero de datos de entrada.
- 2 - Obtención de los centroides de los clusters definidos a partir de la lectura del fichero de datos de entrada.
- 3 - Obtención de los árboles de decisión generados para cada cluster mediante la lectura de los ficheros de datos de entrada, y el posterior análisis y traducción de la versión simplificada que contienen éstos.
- 4 - Reparto de las instancias en los clusters, según su afinidad con los respectivos centroides.

- 5 - Para todos los clusters:
 - 5.1 - Clasificación de las instancias pertenecientes al cluster actual, a través del árbol de decisión correspondiente al mismo.
 - 5.2 - Cálculo e impresión de las métricas de evaluación del cluster concreto.
 - 5.3 - Si se ha realizado para todos los clusters, continuar al siguiente paso. Si no, ejecutar desde el paso 5.1, con el siguiente cluster.
- 6 - Cálculo e impresión de las métricas de evaluación de la clasificación total.

3.3 Diagramas de clases

Una vez expuesto el funcionamiento de los módulos, se ha procedido a determinar la estructura de cada uno, y a modelarla, de cara a la implementación. De nuevo, por cada módulo se muestra un diagrama, en este caso, de clases, en el que se pueden consultar los atributos y métodos de cada clase, así como las relaciones existentes entre las diferentes clases.

Hay que apuntar que para una mejor visualización y comprensión de los diagramas, se ha hecho uso de etiquetas referenciando varios elementos a la vez, como es el caso de los parámetros, y de algunos métodos que, por su carácter auxiliar, no se ha considerado imprescindible incluir de forma específica en los diagramas.

Módulo 1: PSO

La estructura del módulo de clustering con el enjambre de partículas, contempla, básicamente los elementos del enjambre, adaptados para integrar los aspectos necesarios de los clusters a generar.

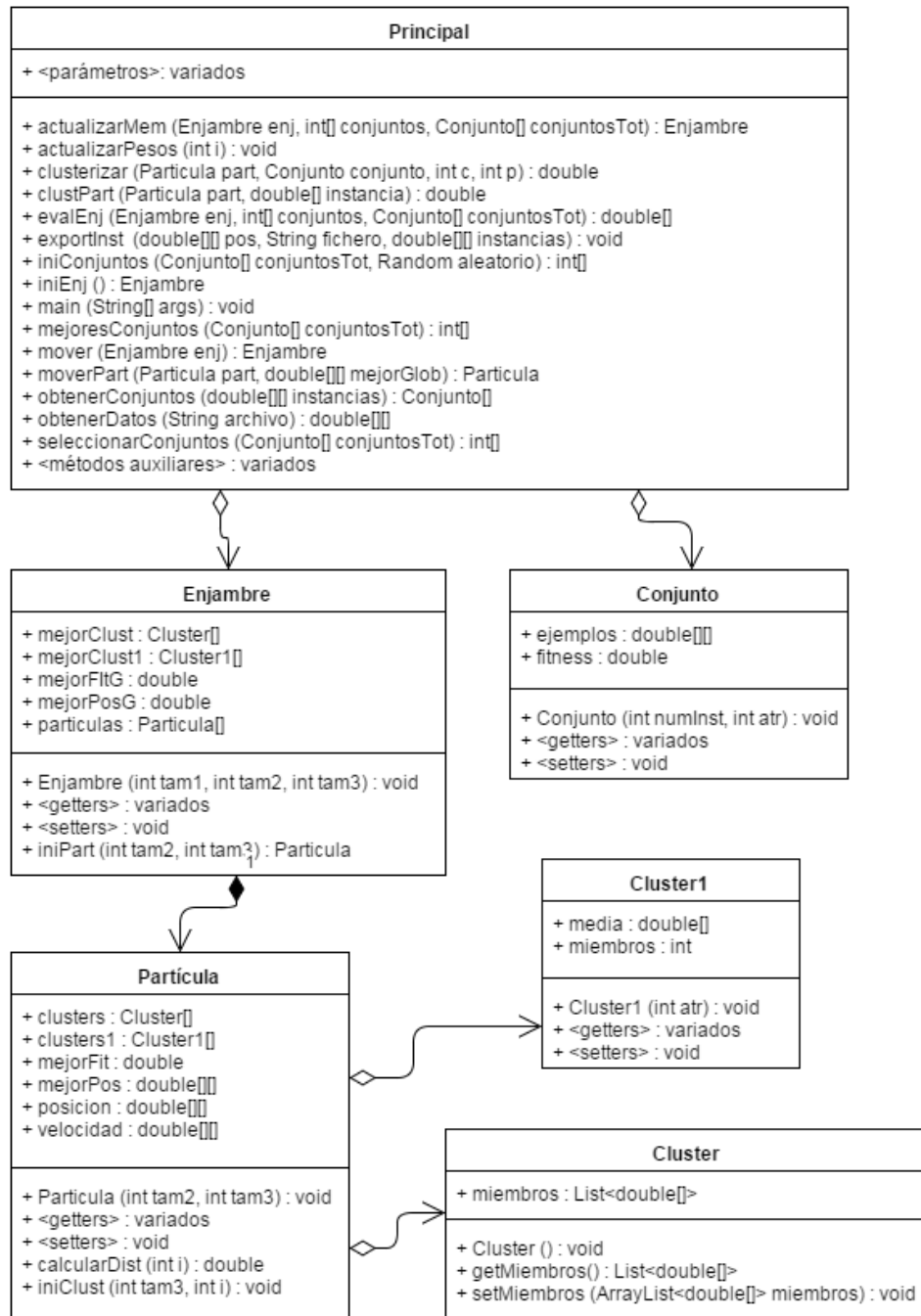


Ilustración 6: Diagrama de clases Módulo 1: PSO

Módulo 2: AE

El modelo de clases definido para el algoritmo evolutivo, refleja los elementos necesarios para la codificación de las soluciones, en este caso, los árboles, que componen la población, para la aplicación de los operadores genéticos y la evaluación de los individuos en base a la clasificación que realicen de los ejemplos de entrenamiento.

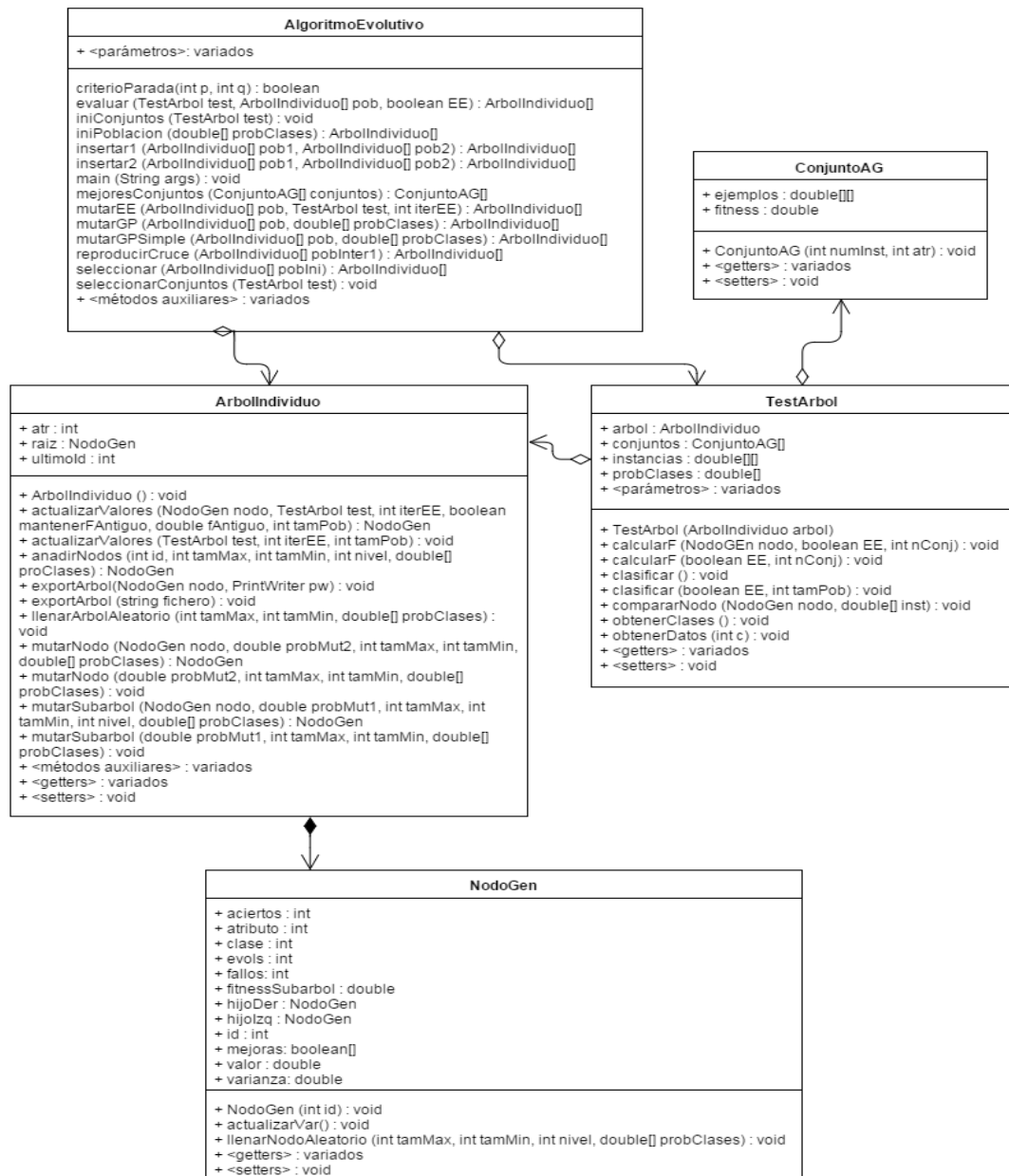


Ilustración 7: Diagrama de clases Módulo 2: AE

Módulo 3: PEv

En el caso del programa evaluador, la estructura fijada corresponde, principalmente, a los productos de los otros módulos en versión simplificada, requeridos para la fase de test o para la clasificación de datos desconocidos.

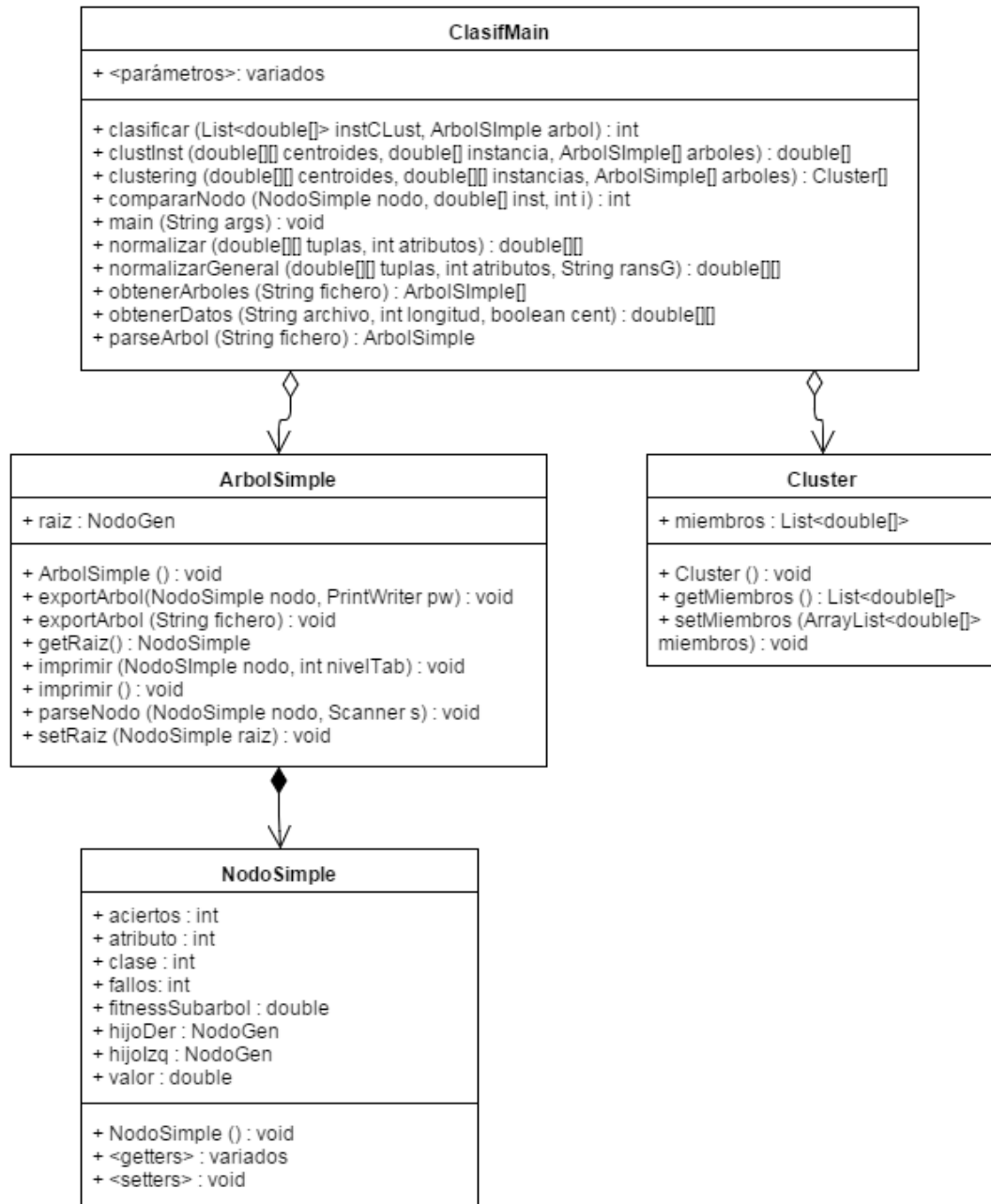


Ilustración 8: Diagrama de clases Módulo 3: PEv

3.4 Implementación

Llegado este punto del trabajo, se tiene una idea del problema que se va a tratar de resolver, los requisitos a cumplir para ello y el diseño del funcionamiento básico y la estructura de la solución propuesta. En la descripción del funcionamiento básico del sistema, se han identificado unos procesos y tareas concretas a realizar por cada uno de los módulos, pero desde un punto de vista a alto nivel. En esta sección, se insiste en la organización del contenido en tres epígrafes equivalentes a los tres módulos en los que se separa el sistema, y se explican los procesos más relevantes involucrados en cada uno de ellos desde un punto de vista más específico.

3.4.1 Optimización de clustering: enjambre de partículas

El cometido del módulo de clustering es facilitar la tarea de clasificación al algoritmo evolutivo, agrupando los ejemplos por afinidad, simplificando así el aprendizaje de los mismos. Para ello, se ha implementado un enjambre de partículas, fácilmente adaptable al problema de agrupación, y, con el fin de comprender mejor la funcionalidad implementada en este módulo, los aspectos y tareas de mayor importancia se describen a continuación.

INICIALIZACIÓN DEL ENJAMBRE

Este proceso implica la inicialización de las partículas que lo componen. Como se ha visto en el estudio de las técnicas utilizadas en este trabajo, las partículas de un enjambre se definen en todo momento por medio de dos elementos principales: su posición y su velocidad.

En el sistema propuesto, la posición de una partícula representa una configuración de clusters, siendo cada coordenada, un centroide correspondiente a un atributo de los datos.

Al haber normalizado los datos, las coordenadas se inicializan aleatoriamente con valores entre 0 y 1. Pero, de manera similar a lo que ocurre en el algoritmo k-medias++, se pretende que cada partícula abarque el máximo espacio de datos posible con sus centroides. Por ello, una nueva

configuración de los mismos es escogida definitivamente con una probabilidad directamente proporcional a la distancia entre esta configuración y las de los clusters ya inicializados.

La velocidad, por su parte, también contiene coordenadas equivalentes a los atributos de los ejemplos, y estos valores se inicializan a 0, de manera que las partículas de inicio ya avancen hacia las mejores soluciones.

La memoria global del enjambre y la memoria local de la partícula inicialmente se encuentran vacías, hasta que se realiza la primera evaluación del enjambre.

INICIALIZACIÓN DE LOS CONJUNTOS

En la inicialización de los conjuntos de ejemplos, necesarios en la parte coevolutiva del algoritmo, se divide la totalidad de las instancias en subconjuntos del mismo tamaño y se dispone de una ventana temporal en la que se insertan al azar los índices de un número fijo reducido de los subconjuntos generados.

EVALUACIÓN DE ENJAMBRE Y CONJUNTOS: FUNCIONES DE FITNESS

Para la cuantificación de la aptitud, o fitness, de las partículas del enjambre, con respecto a los conjuntos de ejemplos, y viceversa, se aplican las siguientes funciones de fitness:

- **Fitness de una partícula:** es la media de los errores obtenidos por los clusters de la misma para todas las instancias de los conjuntos de la ventana. El fitness de un encuentro entre una partícula y una instancia se corresponde con la distancia, en concreto la distancia Euclídea, entre los atributos de la instancia y los centroides del cluster más cercano, de los definidos dentro de la partícula. La función de optimización consiste en la minimización del fitness, persiguiendo así la minimización del error de clustering.
- **Fitness de un conjunto:** es la media de los fitness resultantes de los encuentros de sus instancias y todas las partículas del

enjambre. En este caso, como el objetivo es hacer que el enjambre se vea forzado a evolucionar, es decir, “ponérselo difícil” a las partículas, el fitness se toma como el error de clustering, en negativo. Así, la función de evaluación de los conjuntos minimiza este fitness, lo que equivale a maximizar el error de clustering.

Una vez terminada la evaluación de todos los conjuntos con todas las partículas, se actualiza la memoria del enjambre y sus partículas: si el fitness actual de la partícula es el mejor que ha conseguido, se guarda como tal en su memoria local, y si, además es la mejor encontrada por todo el enjambre, se almacena como tal en la memoria global del mismo.

A su vez, cuando se reparten las instancias en los clusters para obtener el fitness, se computa y almacena la media de las instancias dentro de cada cluster, así como el número total de miembros de dicho cluster.

DESPLAZAMIENTO DE PARTÍCULAS

Este proceso consiste en la modificación de la posición de las partículas, acorde con el vector de velocidad de cada una. Este cálculo se realiza mediante la fórmula estándar del movimiento, que suma las componentes de la velocidad a las de la posición. Además, cada vez que se desplazan las partículas, se actualiza la velocidad mediante una fórmula normalmente basada en tres factores. Sin embargo, inspirándose en el algoritmo k-medias, en la solución propuesta se ha añadido un cuarto factor para acelerar la convergencia: la media de las instancias de cada cluster de la partícula. Ésta se agrega a la fórmula con un peso determinado a semejanza del resto de factores:

$$v(t+1) = \underbrace{w \cdot v(t)}_{\text{Inercia}} + \underbrace{\varphi_1 \cdot r(o, a_1) \cdot (x^{max}(t) - x(t))}_{\text{Memoria individual}} + \underbrace{\varphi_2 \cdot r(o, a_2) \cdot (G(t) - x(t))}_{\text{Memoria colectiva}} + \underbrace{p \cdot m(t)}_{\text{Media}}$$

Otra medida tomada para tratar de mejorar la capacidad de convergencia del algoritmo, es la de controlar que, en las operaciones entre distintas posiciones de las partículas, se emparejen los centroides de los clusters “correspondientes” entre sí, es decir, los más cercanos. Con esta medida se pretende hacer el recorrido más corto entre dos posiciones, operando cada cluster con su cluster homólogo en la otra posición,

evitándose así que dos partículas con la misma configuración de clusters, pero en distinto orden, sean consideradas partículas con distinta posición.

ACTUALIZACIÓN DE PESOS: INERCIA Y MEDIA

Dentro de la fórmula de actualización de la velocidad, las componentes de la inercia y de la media afectan en gran medida a cómo de drástico sea el cambio en la velocidad de la partícula. Esto implica que ambas componentes afecten significativamente a la convergencia del algoritmo. Con la finalidad de adecuar el avance de las partículas al momento de la ejecución del algoritmo, para que al inicio de la misma se fomente la exploración del espacio de soluciones y al final de la ejecución se refuerce la explotación de dicho espacio, se han definido dos funciones de progreso por las que se actualicen los pesos de la inercia y de la media, respectivamente.

Las dos funciones parten de la función de raíz cúbica, la cual se ha adaptado para que la evolución de ambos pesos reproduzca el punto de inflexión de la misma, una vez pasado el impulso inicial en la convergencia del algoritmo, debido al gran margen de mejora del principio.

En el caso del peso de la **inercia**, su función de progreso es decreciente, para que la inercia tome valores altos en las primeras iteraciones, permitiendo la diversidad de las soluciones, y valores bajos en las últimas iteraciones, cuando se haya localizado la zona óptima dentro del espacio de soluciones y el riesgo de convergencia prematura sea reducido.

En cambio, el peso de la **media** aumenta, sigue una función creciente, produciendo valores bajos de la media al comienzo de la ejecución, evitando acelerar demasiado la convergencia y estancarse en algún mínimo local, y valores altos al final, para cuando ya no convenga que las partículas se alejen demasiado de las distribuciones buenas alcanzadas.

Se puede apreciar gráficamente el recorrido que realizan ambos pesos con sus funciones en el apartado **4.2 Configuración de parámetros**.

SELECCIÓN DE LOS CONJUNTOS

En el módulo de clustering con enjambre de partículas, se retiran todos los conjuntos de la ventana temporal, y se seleccionan e insertan los mejores, aquéllos con fitness menor, en la nueva ventana, tantos como indique el tamaño fijado al inicio, pudiendo darse el caso de escogerse algún conjunto que ya estuviera incluido en la ventana antigua. Para dar prioridad a los conjuntos que aún no hayan sido seleccionados, cuando se inicializa la población de conjuntos, todos ellos parten de un fitness negativo muy grande, lejos de los valores factibles para el error por instancia con atributos normalizados.

AGRUPACIÓN FINAL DE INSTANCIAS

Cuando se cumplen las iteraciones máximas a realizar por el enjambre, y se tiene la distribución de centroides definitiva, los valores correspondientes a la misma se exportan en un fichero, necesario para la evaluación del clasificador al completo.

También, se exportan las instancias agrupadas en los clusters generados, para servir como conjuntos de entrenamiento al algoritmo evolutivo, cada cluster por separado.

3.4.2 Aprendizaje de clasificación: algoritmo evolutivo

Este módulo, dedicado a la elaboración de clasificadores capaces de reconocer actividades humanas a partir del aprendizaje de ejemplos mediante un algoritmo evolutivo, es el módulo principal del sistema. La solución propuesta para la clasificación de actividades emplea fundamentalmente la programación genética. Consiste en un algoritmo evolutivo en el que, aparte de los operadores genéticos propios de la programación genética, aplica la técnica de mutación realizada en estrategias evolutivas. Las características del algoritmo se describen a continuación.

CODIFICACIÓN DE SOLUCIONES

Cada individuo de la población representa una solución, en este caso, un clasificador, mediante un árbol de decisión, en el que cada nodo declara una condición con la forma de:

`'atributo' < 'valor continuo'`

Correspondiendo la rama derecha del nodo al caso verdadero, y la rama izquierda al falso. Cada hoja, por su parte, contiene una clase, que se asigna a las instancias que cumplan todas las condiciones que la preceden.

Además, tanto nodos como hojas, aparte de obviar el operador comparativo que no varía, recogen más información relevante para el desarrollo del mismo (su varianza para la mutación de EE, entre otros), aunque no forme parte del clasificador y no se refleje en la codificación. (Véase **ANEXO II. Árboles** para ver un ejemplo de fenotipo de un árbol).

INICIALIZACIÓN DE ÁRBOLES: CON PROBABILIDADES DE CLASES

Los árboles de población se inicializan generándose de la raíz a las hojas, alcanzando un número de niveles acotados por el tamaño máximo y el tamaño mínimo predefinidos. En cada nodo se escoge un atributo al azar, en este caso, de los 23 disponibles, y se le atribuye un valor continuo aleatorio. En cambio, de tratarse de una hoja, se selecciona una clase de entre las posibles, 12 en el problema que atañe a este trabajo, con una probabilidad directamente proporcional a su frecuencia de aparición en los datos de entrada. También se inicializan las varianzas asociadas a los valores continuos de los nodos, con 1/3 para todos los casos.

INICIALIZACIÓN DE LOS CONJUNTOS

Al igual que con el enjambre de partículas, en el algoritmo evolutivo se aplica la coevolución competitiva, pero sigue un procedimiento distinto. En este módulo, no se inicializa una población fija de conjuntos, sino que se generan aleatoriamente tantos conjuntos como quepan en la ventana

temporal, no necesariamente disjuntos, pudiendo solaparse entre ellos, pero del mismo tamaño.

OPERADORES GENÉTICOS

Tras la inicialización de la población de árboles y la ventana de conjuntos, los operadores genéticos que se aplican son:

Selección

La selección se realiza mediante la técnica de torneos, de tantos individuos como se haya indicado previamente, destinándose los individuos ganadores a la reproducción.

Reproducción

Consiste en el cruce de subárboles, es decir, se escogen aleatoriamente dos padres de entre los seleccionados con los torneos. A su vez, se escoge aleatoriamente un nodo de cada padre y se intercambian, generándose dos descendientes: el primer padre con el subárbol del segundo y el segundo padre con el subárbol del primero.

Mutación GP

Se trata de la mutación propia de la programación genética, se aplica según una probabilidad determinada y se ha implementado en sus tres formas: mutaciones terminal y funcional simples, en las que se muta la información contenida en el nodo (atributo, operador, valor) o en la hoja (clase); y mutación de árbol, en la que se muta el subárbol completo.

Mutación EE

Al ser los valores numéricos contenidos en los nodos valores continuos, se ha decidido aplicar para su evolución la mutación de estrategias evolutivas. Para ello, se seleccionan algunos árboles dentro de la población y cada uno se recorre de manera descendente, realizando en cada nodo la mutación del valor numérico contenido, así como de su varianza, tantas veces como se haya indicado para esa ejecución del

algoritmo. De esta manera, se pretende mejorar la eficacia de cada árbol todo lo posible conservando su estructura.

Inserción

En el algoritmo evolutivo desarrollado se utilizan dos tipos de inserción, dadas población 1 (P1) y población 2 (P2):

- **Inserción 1:** si el mejor individuo de la población 1 es mejor que el peor individuo de la población 2, este último es sustituido por el primero.
- **Inserción 2:** el peor individuo de la población 2 es sustituido por el mejor individuo de la población 1, en el caso de que este último sea mejor que el mejor individuo de la población 2.

La inserción 1 se aplica después de la mutación GP entre la población inicial (P1) y la población generada, población intermedia 2 (P2), obteniéndose la población final; mientras que la inserción 2 se aplica después de la mutación EE entre la población final (P1) y la población generada, población final EE (P2).

EVALUACIÓN DE ÁRBOLES Y CONJUNTOS: FUNCIONES DE FITNESS

Tanto para árboles como para conjuntos, la función de evaluación consiste en el cálculo del porcentaje de acierto de los árboles individuo tras clasificar las instancias de cada conjunto de entrenamiento contenido en la ventana temporal. Este valor constituye el fitness de árboles y conjuntos y, por parte de los primeros, se busca maximizarlo, mientras que por parte de los segundos, se busca minimizarlo.

SELECCIÓN DE CONJUNTOS

En este módulo, al igual que en el de clustering con enjambre de partículas, se vacía la ventana temporal, y se seleccionan e insertan los mejores conjuntos, pudiendo escogerse algún conjunto que estuviera incluido anteriormente. A su vez, para priorizar los conjuntos que nunca hayan sido seleccionados, cuando se inicializa la población de conjuntos,

Después de la poda

```
atributo6 < 0.6883239273185245
|
| atributo0 < 0.15580728026670912
| |
| | clase: 1 (16066 correctas)(51 incorrectas)
| | atributo6 < 0.5192801691796022
| | |
| | | clase: 1 (1492 correctas)(0 incorrectas)
| | | clase: 2 (19363 correctas)(35 incorrectas)
| atributo4 < 0.6377765210747529
| |
| | clase: 10 (21156 correctas)(0 incorrectas)
| | clase: 1 (0 correctas)(273 incorrectas)
```

EXPORTACIÓN DE ÁRBOLES CON CODIFICACIÓN SIMPLIFICADA

Con el objetivo de ser importados en el módulo evaluador, los árboles de decisión generados se exportan a un fichero cada uno, con un formato simplificado con respecto al utilizado durante la ejecución del algoritmo y, a su vez, con respecto a la codificación del fenotipo. De esta última, sólo se conserva para cada nodo el número de atributo y el valor a comparar, o si es una hoja, se indica con una C al inicio de la línea, seguida de la clase a asignar. Se eliminan tabulaciones y demás, y se asume que el operador comparativo es siempre “menor que” por lo que, de nuevo, se prescinde de la codificación del mismo. (Véase [ANEXO II. Árboles](#) para ver un ejemplo de codificación simplificada de un árbol)

3.4.3 Evaluación del reconocimiento de actividades

El módulo de evaluación reúne los productos de los otros dos módulos para simular el proceso completo de predicción de actividades para datos de entrada desconocidos. En este caso, para poder analizar el rendimiento del proceso, los datos de entrada sí están etiquetados y se comparan con las predicciones para calcular las medidas de rendimiento pertinentes.

PROCESAMIENTO DE DATOS

Lo primero de todo es el procesamiento de las instancias nuevas, para que tengan la forma y dimensión de las instancias empleadas en la fase de aprendizaje. Para ello, se llevan a cabo tres acciones:

- La **extracción de atributos** relevantes, es decir, los seleccionados durante la etapa de extracción de atributos de la ARC. Ignorando los demás atributos, insignificantes para el clasificador generado.
- La **traducción de las clases** originales a los códigos utilizados durante el proceso de entrenamiento.

AGRUPACIÓN DE INSTANCIAS SEGÚN CENTROIDES IMPORTADOS

Las instancias resultantes se agrupan en los respectivos clusters según su similitud, o distancia Euclídea a los centroides importados.

SÍNTESIS DE ÁRBOLES IMPORTADOS

Este módulo incorpora un *parser* para la lectura de los árboles en la codificación simplificada, anteriormente descrita, y su traducción o conversión a los árboles simples implementados en el evaluador.

CLASIFICACIÓN DE ACTIVIDADES

El proceso de clasificación de actividades es el último eslabón y consiste en la identificación de la clase de cada instancia, mediante el árbol de decisión vinculado al cluster al que pertenece.

PROCESO ALTERNATIVO CON PROBABILIDADES

Además del procedimiento básico descrito, se ha implementado otro alternativo que incluye el uso de la probabilidad de pertenencia de las instancias a cada uno de los clusters.

En lugar de agrupar las instancias en clusters, se calcula la distancia de las mismas a cada uno de los clusters, siendo inversamente proporcional a ésta la probabilidad de pertenencia de cada instancia a los clusters. De esta manera, cuanto más similar es una instancia a los centroides de un cluster, mayor probabilidad tiene de pertenecer al mismo.

Partiendo de este principio, cada cluster da una predicción con su respectivo árbol, y la clase definitiva es aquella que acumule la máxima probabilidad, constituida por la suma de las probabilidades de pertenencia de los diferentes clusters que la seleccionen.

Con este proceso se pretende permitir una predicción más abierta, que favorezca los casos en los que la pertenencia de una instancia a un cluster concreto no esté claramente definida, sin perjudicar los casos en que sí lo esté.

RESULTADOS

El evaluador, a partir de los resultados obtenidos, elabora un análisis de los mismos, imprimiendo unas métricas determinadas, en particular, matriz de confusión, tasa de acierto, precisión y cobertura. Esto lleva implícito el cálculo de verdaderos positivos (TP), falsos positivos (FP), verdaderos negativos (TN) y falsos negativos (FN) de las predicciones hechas en relación a cada actividad.

4. Experimentación

En esta sección, se explica el procedimiento llevado a cabo para la generación del clasificador a partir del sistema implementado, y la evaluación del rendimiento del mismo.

En primer lugar, se describe la metodología seguida, explicando cómo se han realizado los pasos que conforman la cadena de reconocimiento de actividades, o ARC.

Posteriormente, se detalla la configuración de parámetros escogida para la ejecución de las pruebas, con el objetivo de favorecer la convergencia de los algoritmos.

Después, se exponen las distintas pruebas realizadas, con sus respectivos resultados. Se concluye con un resumen y análisis de los resultados obtenidos.

4.1 Metodología

Generalmente, los sistemas de reconocimiento de actividades humanas se rigen por una metodología común, llamada cadena de reconocimiento de actividades, que comprende las etapas de: adquisición de datos, preprocesamiento de los datos, segmentación de los mismos, generación de atributos y aprendizaje o clasificación.

Este apartado se estructura de acuerdo a este procedimiento, explicando cómo se ha aplicado cada etapa en el sistema propuesto en este trabajo.

4.1.1 Adquisición de datos

Persiguiendo el objetivo de extender la utilización de datos comunes entre los distintos estudios del campo del reconocimiento de actividades, para este trabajo se ha utilizado un conjunto de datos disponible públicamente, el PAMAP2, introducido por Reiss [28, 152, 131] con este fin.

La adquisición de las señales de sensores de este conjunto se realizó mediante tres IMUs *wearable*, concretamente Colibri Wireless [29] (véase sección **2.1.1 Monitorización**), colocadas en la muñeca dominante, en el pecho y en el tobillo dominante de los individuos participantes, así como una unidad de frecuencia cardíaca. Participaron 9 individuos, la mayoría siguiendo un protocolo predefinido de ejecución de 12 actividades y, además, pudiendo realizar otras 6 actividades opcionales. El protocolo está constituido por las actividades de: estar tumbado, estar sentado, estar de pie, caminar, correr, ir en bici, marcha nórdica, subir escaleras, bajar escaleras, pasar la aspiradora, planchar y saltar a la comba. Mientras que las actividades opcionales son: ver la televisión, trabajar en el ordenador, conducir un coche, doblar la colada, limpiar la casa y jugar al fútbol.

En la *Tabla 4*, disponible con la colección de datos [131], junto con otra información referente a la misma, se recoge el tiempo, en segundos, que cada individuo ha estado realizando cada actividad.

Los datos incluyen una cantidad de 53 atributos sin contar la clase, 17 por cada IMU, más la frecuencia cardíaca y el momento en el tiempo (*timestamp*). Las señales captadas por las IMUs son:

- El 1: temperatura (°C).
- Del 2 al 4: 3 ejes de aceleración (ms^{-2}), con escala de $\pm 16\text{g}$ y resolución de 13 bits.
- Del 5 al 7: 3 ejes de aceleración (ms^{-2}), con escala de $\pm 6\text{g}$ y resolución de 13 bits.
- Del 8 al 10: 3 ejes de giroscopio (rad/s).
- Del 11 al 13: 3 ejes de magnetómetro (μT).
- Del 14 al 17: orientación. Datos inválidos para esta colección, según los autores [131].

Tabla 4: Tiempo de realización de actividades

	Sujeto1	Sujeto2	Sujeto3	Sujeto4	Sujeto5	Sujeto6	Sujeto7	Sujeto8	Sujeto9	Sumatorio	Nº sujetos
Estar tumbado	271.86	234.29	220.43	230.46	236.98	233.39	256.1	241.64	0,00	1.925,15	8
Estar sentado	234.79	223.44	287.6	254.91	268.63	230.4	122.81	229.22	0,00	1.851,80	8
Estar de pie	217.16	255.75	205.32	247.05	221.31	243.55	257.5	251.59	0,00	1.899,23	8
Caminar	222.52	325.32	290.35	319.31	320.32	257.2	337.19	315.32	0,00	2.387,53	8
Correr	212.64	92.37	0,00	0,00	246.45	228.24	36.91	165.31	0,00	981.92	6
Ir en bici	235.74	251.07	0,00	226.98	245.76	204.85	226.79	254.74	0,00	1.645,93	7
Marcha nórdica	202.64	297.38	0,00	275.32	262.7	266.85	287.24	288.87	0,00	1.881,00	7
Ver la televisión	836.45	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	836.45	1
Trabajar en el ordenador	0,00	0,00	0,00	0,00	1108.82	617.76	0,00	687.24	685.49	3.099,31	4
Conducir un coche	545.18	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	545.18	1
Subir escaleras	158.88	173.4	103.87	166.92	142.79	132.89	176.44	116.81	0,00	1.172,00	8
Bajar escaleras	148.97	152.11	152.72	142.83	127.25	112.7	116.16	96.53	0,00	1.049,27	8
Pasar la aspiradora	229.4	206.82	203.24	200.36	244.44	210.77	215.51	242.91	0,00	1.753,45	8
Planchar	235.72	288.79	279.74	249.94	330.33	377.43	294.98	329.89	0,00	2.386,82	8
Doblar la colada	271.13	0,00	0,00	0,00	0,00	217.85	0,00	236.49	273.27	998.74	4
Limpiar la casa	540.88	0,00	0,00	0,00	284.87	287.13	0,00	416.9	342.05	1.871,83	5
Jugar al fútbol	0,00	0,00	0,00	0,00	0,00	0,00	0,00	181.24	287.88	469.12	2
Saltar a la com ba	129.11	132.61	0,00	0,00	77.32	2.55	0,00	88.05	63.9	493.54	6
Total etiquetado	4.693,07	2.633,35	1.743,27	2.314,08	4.117,97	3.623,56	2.327,63	4.142,75	1.652,59	27.248,27	

Sin embargo, no todos los sensores tiene la misma frecuencia de muestreo, las IMUs tienen una frecuencia de 100Hz, mientras que el monitor de frecuencia cardíaca capta señal con una frecuencia de 9Hz. Como consecuencia, en el 91% de la colección de datos se desconoce la frecuencia cardíaca.

4.1.2 Preprocesamiento de datos

Previamente a la segmentación de los datos y la extracción de atributos, ha sido necesario preparar el conjunto de datos, considerando aspectos como valores desconocidos, datos erróneos y datos irrelevantes o ruido.

Primeramente, se ha decidido tomar en cuenta únicamente los datos correspondientes al protocolo de actividades estándar, manteniéndose sólo los datos de esas 12 actividades (estar tumbado, estar sentado, estar de

pie, caminar, correr, ir en bici, marcha nórdica, subir escaleras, bajar escaleras, pasar la aspiradora, planchar y saltar a la comba). Consecuentemente, se ha retirado los datos referentes al sujeto 9, ya que se apartan demasiado del protocolo definido. También, con el fin de evitar el sesgo de los datos, ha sido necesario eliminar el atributo del *timestamp*, ya que las actividades del protocolo han sido realizadas con una duración y orden específicos e iguales para todos los individuos. Por error, este ajuste no se aplicó hasta ya empezada la tarea de Experimentación, afectando a la planificación final del proyecto (véase sección

5.3 Planificación final **5. Planificación**).

Además, en este trabajo se han llevado a cabo las siguientes operaciones:

- **Valores desconocidos.** Los valores perdidos por fallos de comunicación representan una parte muy pequeña de los datos, en torno al 1% de algunos atributos, por lo que se ha eliminado las instancias que contienen estos valores *NaN*. En cambio, los valores desconocidos por la baja frecuencia de muestreo de la unidad de ritmo cardíaco, constituyen una proporción muy alta de los datos, en torno al 91%. Considerando que entre un valor conocido y el siguiente, cada 0.01 segundos, la frecuencia cardíaca no varía, se ha sustituido cada *NaN* por el último valor conocido del atributo.
- **Datos erróneos.** Puesto que los autores de la colección indican que los datos de orientación no son válidos, se han retirado. Por otro lado, los datos del segundo acelerómetro son menos precisos que los del primer acelerómetro, y siguiendo la recomendación de los autores, sólo se han utilizado los datos del primer acelerómetro.
- **Datos irrelevantes y ruido.** En los datos se incluyen las instancias recogidas entre una actividad y otra, clasificadas como instancias de transición, que no representan ninguna de las actividades que se pretende aprender a reconocer, por lo que se han eliminado.

4.1.3 Segmentación de los datos

La etapa de segmentación consiste comúnmente en la agrupación de los datos en ventanas que abarcan las instancias recogidas en el transcurso de cierto tiempo. Muchos de los trabajos de reconocimiento de actividades, establecen la ventana temporal en una duración en torno a los 5 o 6 segundos, de manera que no sea excesivo el retardo de la predicción en tiempo real, y se disminuya considerablemente el número de instancias, no así su dimensión, sin perder información.

Sin embargo, en este trabajo se ha decidido no segmentar los datos con ninguno de los procedimientos convencionales, tratando de suplir esta fase alternativamente con la fase de clustering, cuyo algoritmo se ha descrito anteriormente (véase sección **3.4 Implementación**). De este modo, se presenta un nuevo enfoque, por el cual cada instancia se clasifica de manera independiente a las demás, lográndose un menor retardo en la respuesta del clasificador. Por otro lado, para evitar una ejecución de la

clasificación demasiado frecuente, se podría clasificar una instancia independiente cada cierto tiempo, suficientemente corto como para no perderse ninguna actividad, pudiéndose prescindir de las instancias intermedias y permitiéndose una reducción en la frecuencia de muestreo de los sensores.

4.1.4 Selección de atributos

Tras la fase de preprocesamiento, los datos cuentan con 31 atributos, de entre los cuales deben seleccionarse los más relevantes. Para evitar la pérdida de información posiblemente relevante para la fase de clustering, la selección de atributos se ha llevado a cabo posteriormente, es decir, justo antes de la fase de clasificación, una vez obtenidos los clusters correspondientes. En el contexto de clasificación, los atributos más relevantes serán los que mejor separen las instancias en clases. Esta tarea se ha realizado mediante la herramienta de aprendizaje automático Weka [229, 230], aplicando el selector de atributos que evalúa subconjuntos de atributos mediante búsqueda genética, cuya aptitud corresponde a la tasa de acierto obtenida por el clasificador Random.

Con este método, se ha obtenido una selección de atributos distinta para cada sujeto, utilizándose en la clasificación dependiente de cada uno, respectivamente. En el caso de la clasificación independiente y parcialmente dependiente del sujeto, se ha llevado a cabo un proceso de votación, escogiéndose dos combinaciones de atributos seleccionados por un número determinado de sujetos. Quedando las configuraciones de atributos de la siguiente manera:

- **Clasificación dependiente del sujeto:**
 - *Sujeto 1:* 9 atributos.
 - *Sujeto 2:* 8 atributos.
 - *Sujeto 3:* 16 atributos.
 - *Sujeto 4:* 16 atributos.
 - *Sujeto 5:* 14 atributos.
 - *Sujeto 6:* 13 atributos.
 - *Sujeto 7:* 11 atributos.
 - *Sujeto 8:* 16 atributos.
- **Clasificación independiente y parcialmente dependiente del sujeto:**
 - *Votados al menos por 4 sujetos:* 11 atributos.
 - *Votados al menos por 3 sujetos:* 22 atributos.

4.1.5 Entrenamiento y clasificación

En la etapa final hay, a su vez, dos modalidades: la de entrenamiento, cuando se genera el clasificador aplicando el algoritmo de aprendizaje, y la de clasificación, cuando se pone a prueba el clasificador generado. Si se está en modo de entrenamiento, o fase de aprendizaje, los datos sometidos a las etapas anteriores de la ARC son los correspondientes al conjunto de entrenamiento. Por el contrario, si el sistema se encuentra en modo de clasificación, que incluye la evaluación de rendimiento, los pasos previos se aplicarán a los datos de test.

Por lo tanto, la tarea a realizar de primera mano es la elaboración de conjuntos de entrenamiento y de test adecuados, primordial para que el aprendizaje sea exitoso. En este caso, como ya se ha visto, se aplica la técnica de coevolución, de manera que no es imprescindible definir un único conjunto de entrenamiento ideal, y simplemente se han creado los conjuntos de acuerdo a los distintos experimentos a ejecutar:

- ❖ **Clasificación independiente del sujeto.** Se han creado 8 conjuntos de test, cada uno coincidente con el conjunto de datos de solamente un sujeto, y 8 conjuntos de entrenamiento, cada uno conformado conjuntamente por los otros 7 sujetos no incluidos en conjunto de test. Además, los datos de los conjuntos han sido normalizados en base a unos rangos generales extraídos del total de los datos disponibles.
- ❖ **Clasificación dependiente del sujeto.** Cada sujeto se ha dividido en 10 partes, a partir de las cuales se han generado 10 conjuntos de test correspondientes a cada parte, y otros 10 conjuntos de entrenamiento, cada uno compuesto por las 9 partes excluidas del conjunto de test correspondiente. Al tratarse los sujetos como problemas diferentes, los datos de cada uno han sido normalizados de manera aislada.
- ❖ **Clasificación parcialmente dependiente del sujeto.** El conjunto de los 8 sujetos mezclados se ha dividido en 10 partes, formándose de nuevo 10 conjuntos de test, uno por cada parte, y 10 conjuntos de entrenamiento, cada uno con las 9 partes restantes, en cada caso. Para la normalización de estos conjuntos se ha seguido el mismo procedimiento que en la clasificación independiente del sujeto.

En los siguientes apartados se explica en profundidad la etapa de entrenamiento y de clasificación, detallándose la configuración de

parámetros establecida para la fase de aprendizaje de cada experimento, así como la evaluación de clasificación efectuada sobre los clasificadores generados.

4.2 Configuración de parámetros

Durante la fase de entrenamiento, se han definido los valores que deben tomar los parámetros de los dos algoritmos de aprendizaje, tanto el enjambre de partículas como el algoritmo evolutivo, mediante experimentos de pequeña dimensión. Básicamente, los parámetros son resultado de buscar la mejora de la convergencia de los algoritmos, con el método de prueba y error, partiendo de valores, intuitivamente y en teoría, válidos.

4.2.1 Enjambre de partículas

En el caso del enjambre de partículas, la mayoría de los parámetros definidos se recogen en la *Tabla 5*, mientras que los parámetros correspondientes a los pesos se tratan aparte.

Tabla 5: Enjambre de partículas

Parámetro	Valor
Iteraciones intermedias	10
Iteraciones completas	50
Partículas	25
Clusters	5
Conjuntos por ventana	5
Proporción de instancias por conjunto	0.05
Conjuntos totales	1/proporción de instancias por conjunto

Las **iteraciones intermedias** implican el número de veces que se ejecuta el algoritmo sin variar la ventana de conjuntos de ejemplos. Respecto a los experimentos a realizar, el parámetro determinado permite que el enjambre se adapte a los conjuntos de la ventana, sin especializarse demasiado.

Las **iteraciones completas** son utilizadas como criterio de parada. Se ha observado que, generalmente, al cabo de ese número de iteraciones, la convergencia del algoritmo no es significativa.

El **número de partículas** del enjambre establecido mantiene un tiempo de ejecución aceptable sin comprometer la convergencia de manera notable.

De igual forma, el **número de clusters** predeterminado permite obtener una agrupación útil de los ejemplos, ya que, habiendo 12 actividades, un valor más alto supondría prácticamente la especialización de cada cluster en una actividad, limitándose o, incluso, anulándose la labor del algoritmo evolutivo.

La **proporción de instancias por conjunto**, indicada respecto al conjunto de entrenamiento, otorga un balance entre el tiempo de ejecución de los experimentos más largos y la capacidad de generalización del algoritmo.

Los **conjuntos totales** sencillamente son el valor resultante de dividir la unidad entre la proporción de instancias por conjunto, equivalente a las partes en que hay que dividir el total del conjunto de entrenamiento para tener esa proporción de instancias en cada conjunto reducido.

Por otra parte, la combinación de valores fijada para los pesos, parámetros esenciales en la optimización con enjambres de partículas, se presenta en la *Tabla 6*.

Tabla 6: Combinación de valores

Factor	Peso
Memoria individual	0.3
Memoria global	0.18
Inercia	0.08 – 0.16 (min – max)
Media	0.12 – 0.22 (min – max)

Al tratarse de los factores que modifican la velocidad de las partículas, con esos pesos se pretende equilibrar la importancia de unos y otros, además de que el total de los pesos no sume más de 1, pues es el límite superior de la velocidad en un entorno con valores normalizados.

Por un lado, entre la **memoria individual** y la **global**, se ha dado más peso a la memoria individual para mantener la diversidad de las partículas y que tiendan a desplazarse hacia la mejor posición global exclusivamente.

Por otro lado, la **inercia** se ha ajustado de manera que en un comienzo fuerce a las partículas a la exploración, no sólo guiadas por los mejores resultados hallados. Para ello, se ha determinado una función decreciente que rige la evolución del peso de la inercia, y que corresponde a la raíz cúbica, pero deformada para satisfacer los valores mínimo y máximo. La evolución del peso de la inercia a lo largo de las iteraciones definidas se muestra en la siguiente gráfica:

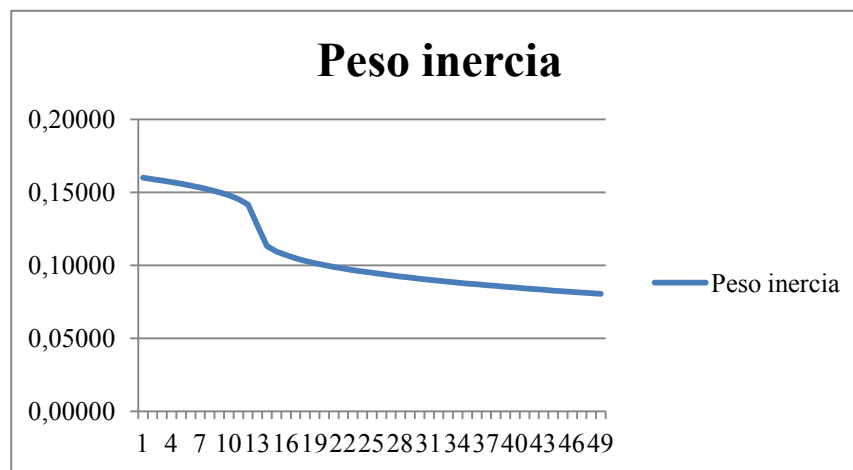


Ilustración 9: Peso inercia

La **media**, por su parte, como ya se ha mencionado, no es un factor propio del movimiento del enjambre de partículas, sino que se ha agregado para favorecer la convergencia, inspirándose en el algoritmo k-medias. A semejanza de la inercia, el peso de la media evoluciona de acuerdo a una función alterada de la raíz cúbica, pero creciente en este caso. Así, se pretende minimizar la función de fitness, es decir, la distancia media de las instancias a los centroides de sus respectivos clusters, haciendo especial hincapié en las iteraciones finales, cuando la distribución de los clusters está más definida y no se provoca una convergencia prematura. La curva descrita por el peso de la media en el transcurso de las iteraciones se puede observar en la *Ilustración 10*:

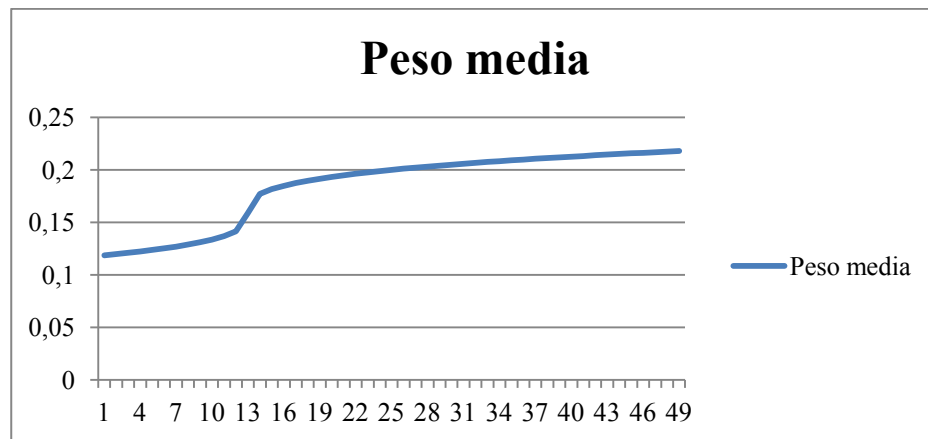


Ilustración 10: Peso media

4.2.2 Algoritmo evolutivo

De la misma manera que con el enjambre de partículas, los valores determinados para los distintos parámetros involucrados en la ejecución del algoritmo evolutivo se muestran en la *Tabla 7*, seguida de una explicación breve de cada uno.

Tabla 7: Algoritmo evolutivo

Parámetro	Valor
Tamaño inicial de cada árbol	2 – 4 (min – max)
Árboles	200
Tamaño de torneos	30
Iteraciones completas	100
Probabilidad de mutación de árbol	0.01
Probabilidad de mutación simple	0.02
Iteraciones de estrategias evolutivas	100
Proporción de fitness para estrategias evolutivas	0.05
Conjuntos por ventana	5
Proporción de instancias por conjunto	0.01

El **tamaño inicial de cada árbol** se ha establecido entre los 2 y los 4 niveles, de manera que de inicio los árboles creados aleatoriamente no sean demasiado grandes y crezcan a medida que se adapten al conjunto de datos de entrenamiento.

La población está formada por un **número de árboles** individuos lo suficientemente grande como para abarcar una parte significativa del espacio de soluciones, sin causar un tiempo de ejecución demasiado largo.

El **tamaño de torneos** se ha definido acorde con el tamaño de la población, para que el operador de selección no implique poblar la misma con súper-individuos y se pueda mantener la diversidad, a la par que favorecer la evolución de los individuos más prometedores.

Las **iteraciones completas** que se ejecuta el algoritmo marcan uno de los criterios de parada, considerándose un tope adecuado en consideración al tiempo limitado del que se dispone.

Tanto la **probabilidad de mutación de árbol** como la **probabilidad de mutación simple**, se han ajustado teniendo en cuenta la convergencia del algoritmo, fomentando la diversidad en la población de árboles y, por tanto, contribuyendo a impedir el estancamiento en mínimos locales.

El **número de iteraciones** que se ejecutan las estrategias evolutivas dentro de cada iteración completa, se ha observado ser suficientemente grande como para que las estrategias evolutivas supongan un avance positivo en el algoritmo, pero controlando su incidencia en el crecimiento del tiempo de ejecución.

De un modo similar, la **proporción** que determina el rango de **mejores árboles** a los que se van a aplicar las estrategias evolutivas, promueve la evolución de los mejores árboles, reduciendo el tiempo de ejecución pero tratando de no ser demasiado elitista.

Respecto a los **conjuntos por ventana**, se reincide en la misma cantidad definida para el enjambre de partículas.

A su vez, la **proporción de instancias por conjunto** se ha establecido, al igual que para el enjambre de partículas, para obtener un balance entre el tiempo de ejecución de los experimentos más largos y la capacidad de generalización del algoritmo.

4.3 Experimentos

Para evaluar con rigor el rendimiento del sistema, se han ejecutado varios experimentos correspondientes a diferentes posibles enfoques de aplicación del clasificador: independiente del sujeto, dependiente del sujeto y parcialmente dependiente del sujeto.

Los tres tipos de experimentos conllevan una validación cruzada con los conjuntos de entrenamiento y de test creados para este efecto. Como se ha explicado en la sección [4.1 Metodología](#), estos conjuntos son específicos para cada tipo de experimento.

En todos los experimentos se mide el rendimiento de la clasificación conseguida mediante tres métricas distintas, en concreto, la tasa de acierto (*accuracy*), la precisión (*precision*) y la cobertura (*recall*), referentes a la recuperación de la información, además de la matriz de confusión correspondiente.

En una matriz de confusión se agrupan las distintas predicciones que afectan a cada clase, con respecto a los datos reales, siendo para una clase C1: verdaderos positivos (*true positive, TP*), se predice la clase C1 y la etiqueta real coincide; verdaderos negativos (*true negative, TN*), se predice otra clase y la etiqueta real coincide; falsos positivos (*false positive, FP*), se predice la clase C1 pero la etiqueta real contiene otra clase; y falsos negativos (*false negative, FN*), se predice otra clase pero la etiqueta real contiene la clase C1. La *Tabla 8* refleja estos conceptos, representándose a la izquierda las clases de las etiquetas y arriba, las clases de las predicciones.

Tabla 8: Matriz de confusión

	C1'	Otra clase'
C1	TP	FN
Otra clase	FP	TN

Por su parte, el cálculo de las métricas mencionadas se realiza mediante las fórmulas siguientes:

❖ Tasa de acierto:

$$accuracy = \text{instancias bien clasificadas} / \text{total de instancias}$$

❖ Precisión:

$$precision = TP / (TP + FP)$$

❖ Cobertura:

$$recall = TP / (TP + FN)$$

A partir de estas medidas se puede hacer un análisis más minucioso de varios aspectos de la clasificación producida. A continuación, se exponen las pruebas realizadas y los resultados obtenidos, en base a dichas métricas.

4.3.1 Experimentos independientes del sujeto

Para evaluar la capacidad del sistema de reconocer actividades realizadas por un sujeto desconocido hasta el momento, se ha llevado a cabo una validación cruzada inter-sujeto, es decir, se entrena un modelo con los datos de todos los sujetos menos uno, y se hace la clasificación de actividades del sujeto restante. Este procedimiento se repite tantas veces como sujetos, ocho en el caso que ocupa a este trabajo, tomando como conjunto de test un sujeto distinto cada vez y los otros siete como conjunto de entrenamiento.

El rendimiento de los clasificadores generados se expone por separado para cada sujeto de test, con la respectiva matriz de confusión y una tabla donde se recogen las métricas descritas previamente.

❖ Con 11 atributos

➤ Sujeto 1:

Procedimiento básico

Tabla 9: Matriz de confusión

	Predicciones											
	0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	24466	14999	0	0	0	0	0	0	3	0	0
	1	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	14180	13803	4105	17136	5226	0	0	663	879
	4	0	0	0	0	3684	0	0	6385	2622	0	0
	5	2713	0	0	0	0	0	3342	4808	4881	0	0
	6	0	0	0	0	12877	0	0	0	0	0	11345
	7	0	0	0	0	0	10617	3801	7462	14508	0	0
	8	0	3052	0	5311	0	6356	3466	0	0	0	0
	9	0	0	0	2238	0	0	744	2018	0	3134	0
	10	0	5414	7413	31	0	0	0	329	0	349	22830
	11	0	0	0	0	18	0	0	0	0	0	0

Tabla 10: tabla de clases con precisión y cobertura

Clase	Precisión	Cobertura
0	0,479424675	0,619894598
1	-	-
2	-	-
3	0,341920781	0,24651736
4	0,1217127	0,290284454
5	0	0
6	0	0
7	0,125168769	0,104457513
8	0	0
9	0,105521886	0,385296287
10	0,462183173	0,627784194
11	0	0

Tasa de acierto total: **29,0111%**

➤ **Sujeto 2:**

Procedimiento básico

Tabla 11: Matriz de confusión

		Predicciones											
		0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	21487	19957	1	0	8532	0	0	0	17	0	0	1350
	1	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	84	0	6	0
	3	795	0	0	24553	0	0	0	4189	2485	0	0	0
	4	0	0	0	0	0	2360	36	0	0	0	0	0
	5	0	0	0	6947	0	15077	27980	0	0	0	0	0
	6	0	0	0	0	0	6885	435	0	0	0	0	0
	7	0	0	0	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	7564	4935	0	0	0
	9	41	1699	24522	0	0	0	0	5477	4674	20665	1553	0
	10	949	666	893	0	0	0	0	104	2995	0	27248	0
	11	0	0	0	0	243	273	0	0	0	0	0	11451

Tabla 12: tabla de clases con precisión y cobertura

Clase	Precisión	Cobertura
0	0,292989896	0,418490963
1	-	-
2	0	0
3	0,321361727	0,766754107
4	0	0
5	0,224360119	0,301515879
6	0,008319786	0,05942623
7	-	-
8	0,086906754	0,394831587
9	0,284975522	0,352458597
10	0,344484058	0,829341044
11	0,180897616	0,956881424

Tasa de acierto total: **48,5671%**

➤ **Sujeto 3:**

Procedimiento básico

Tabla 13: Matriz de confusión

	Predicciones											
	0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	18113	0	0	0	0	0	0	961	5330	0	0
	1	3883	24242	0	0	0	0	0	93	1041	851	0
	2	45	4496	20408	0	0	0	21	1263	6267	14802	0
	3	0	0	0	4958	0	0	9625	8556	1729	0	0
	4	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	726	0	0	634	4395	0	0	0
	6	0	0	0	22609	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	86	0	4169	2393	0
	9	0	0	0	0	0	0	0	0	495	0	0
	10	0	0	0	0	0	0	0	0	1257	9893	0
	11	0	0	0	0	0	0	0	0	0	0	0

Tabla 14: tabla de clases con precisión y cobertura

Clase	Precisión	Cobertura
0	0,812898304	0,742214391
1	0,853261061	0,80511458
2	0,830369858	0,431440531
3	0,543223403	0,199372688
4	-	-
5	0	0
6	0	0
7	-	-
8	0	0
9	0,106132075	1
10	0,703527237	0,887264574
11	-	-

Tasa de acierto total: **45,0608%**

➤ **Sujeto 4:**

Procedimiento básico

Tabla 15: Matriz de confusión

	Predicciones											
	0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	19968	0	4	5	0	0	0	0	0	0	0
	1	70	20626	683	70	0	0	0	0	0	0	0
	2	545	2748	609	0	0	0	0	51	0	0	0
	3	1935	1099	0	16406	0	0	0	6450	3847	0	197
	4	0	0	0	4447	1	12089	0	4698	3641	0	0
	5	442	0	0	2583	0	10520	26690	1642	4400	0	0
	6	0	0	0	6494	0	0	0	2049	0	0	0
	7	0	0	2483	0	0	0	0	135	700	0	14189
	8	0	0	0	5	0	0	0	1709	1640	19972	9115
	9	69	1000	20613	0	0	0	0	0	0	0	0
	10	0	5	259	20	0	0	0	0	0	21	1441
	11	0	0	0	0	0	0	0	0	0	0	0

Tabla 16: tabla de clases con precisión y cobertura

Clase	Precisión	Cobertura
0	0,243064601	0,999549482
1	0,249079206	0,961629913
2	0,009698688	0,154060207
3	0,208756951	0,548072426
4	1,61E-05	4,02E-05
5	0,144698293	0,22732675
6	0	0
7	0,002166308	0,007711201
8	0,025696066	0,050553312
9	0	0
10	0,022648686	0,825315006
11	-	-

Tasa de acierto total: **31,2393%**

➤ **Sujeto 5:**

Procedimiento básico

Tabla 17: Matriz de confusión

	Predicciones											
	0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	20687	560	0	0	0	0	0	0	2	0	0
	1	499	20810	1006	0	0	0	131	0	4022	7	0
	2	321	2153	7285	0	0	0	77	0	1335	452	0
	3	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	7236	0	0	21743	0	0	0	756
	7	0	0	0	21232	23847	23667	0	10067	7870	138	0
	8	0	0	0	0	0	0	0	0	0	0	0
	9	1341	69	2839	2004	443	194	2166	2655	3051	12698	2054
	10	822	3128	10941	81	11	500	2023	1320	1794	6201	30381
	11	0	0	0	0	0	0	0	0	0	0	0

Tabla 18: tabla de clases con precisión y cobertura

Clase	Precisión	Cobertura
0	0,642912639	0,973551697
1	0,644272446	0,786024551
2	0,388015979	0,626774499
3	-	-
4	-	-
5	-	-
6	0,654259321	0,731225828
7	0,46699448	0,107653481
8	-	-
9	0,52497106	0,430236498
10	0,725585728	0,531117793
11	-	-

Tasa de acierto total: **45,9212%**

➤ **Sujeto 6:**

Procedimiento básico

Tabla 19: Matriz de confusión

		Predicciones											
		0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	8672	0	0	745	0	1633	202	0	35	77	0	0
	1	5	0	48	0	0	0	0	28	918	1511	8	0
	2	0	0	0	0	0	0	0	46	375	31	2096	0
	3	0	0	0	1672	0	0	13907	1828	368	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	22191	9831	0	0	1	0	0	253
	6	0	0	0	0	0	8968	8373	650	2392	0	0	0
	7	0	0	0	9	0	0	0	2934	2695	0	0	0
	8	0	0	0	0	0	0	0	3283	323	0	0	0
	9	397	330	7009	0	0	0	0	462	325	6804	5752	0
	10	14256	22478	17007	22930	0	0	4008	4054	3826	12606	29728	0
	11	0	0	0	0	0	0	0	0	0	0	0	0

Tabla 20: tabla de clases con precisión y cobertura

Clase	Precisión	Cobertura
0	0,48672616	0,76311158
1	0	0
2	0	0
3	0,154571508	0,094064698
4	-	-
5	0,518075464	0,304591647
6	0,477965521	0,410783496
7	0,242900902	0,520397304
8	0,034114913	0,089572934
9	0,426609819	0,322785711
10	0,764746739	0,227116805
11	-	-

Tasa de acierto total: **27,5463%**

➤ **Sujeto 7:**

Procedimiento básico

Tabla 21: Matriz de confusión

		Predicciones											
		0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	25288	0	22	43	0	0	52	730	147	1306	15146	0
	1	66	12216	15957	2035	0	0	500	2128	2672	2588	4595	0
	2	0	0	9733	229	1196	0	692	0	0	0	0	0
	3	0	0	0	24328	1136	20308	14318	3934	445	0	0	0
	4	256	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0	0	0	0	0	0
	7	0	0	0	2	586	1286	8082	5143	2031	0	0	0
	8	0	0	0	0	577	989	3974	3328	3197	0	0	0
	9	0	0	0	266	0	0	0	669	262	5656	0	0
	10	0	0	0	5348	0	0	790	1690	2858	11960	9617	0
	11	0	0	0	0	0	0	0	0	0	0	0	0

Tabla 22: tabla de clases con precisión y cobertura

Clase	Precisión	Cobertura
0	0,566474765	0,591753639
1	0,386961893	0,285707603
2	0,334628344	0,821350211
3	0,556946956	0,377359661
4	0	0
5	-	-
6	-	-
7	0,209952645	0,300233508
8	0,141773836	0,264981351
9	0,226158583	0,825331971
10	0,331964101	0,298081394
11	-	-

Tasa de acierto total: **41,314%**

➤ **Sujeto 8:**

Procedimiento básico

Tabla 23: Matriz de confusión

		Predicciones											
		0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	0	0	22	0	0	0	0	119	0	90	25827	0
	1	2154	19237	15755	0	0	0	0	0	0	1839	0	0
	2	10	3661	3372	0	0	0	0	0	0	2906	0	0
	3	0	0	0	24051	0	9795	0	5821	4271	0	0	0
	4	0	0	0	91	16194	0	28195	1047	4552	0	0	8467
	5	0	0	0	713	0	6312	225	507	0	17659	0	0
	6	0	0	0	0	0	9123	0	0	0	0	0	0
	7	0	0	0	5488	0	0	0	2546	318	0	0	0
	8	0	0	0	0	0	0	0	54	0	0	0	0
	9	1040	0	87	0	0	0	0	1401	510	17	5201	0
	10	20955	0	5833	0	0	0	0	184	0	1709	1857	0
	11	0	0	0	0	0	0	0	0	0	0	0	0

Tabla 24: tabla de clases con precisión y cobertura

Clase	Precisión	Cobertura
0	0	0
1	0,47223586	0,493446197
2	0,135579591	0,338928536
3	0,528013172	0,547384952
4	0,429628844	0,276603013
5	0,226960555	0,248347498
6	0	0
7	0,105884799	0,304837165
8	0	0
9	7,90E-04	0,002059109
10	0,079508477	0,060809483
11	-	-

Tasa de acierto total: **28,388%**

➤ Todos los sujetos:

Tabla 25: tasa de acierto de los sujetos y media por procedimiento básico

Sujeto	Tasa de acierto (%)
1	29,0111
2	48,5671
3	45,0608
4	31,2393
5	45,9212
6	27,5463
7	41,3140
8	28,3880
Media	37,13

Tabla 26: tasa de acierto de los sujetos y media por procedimiento con probabilidades de pertenencia

Sujeto	Tasa de acierto (%)
1	29,1390
2	8,6142
3	43,5488
4	31,0917
5	16,1553
6	34,3921
7	1,5170
8	18,7994
Media	22,91

Tasa de acierto de mejor sujeto → **48,5671 %**

❖ Con 22 atributos

➤ Sujeto 1:

Procedimiento básico

Tabla 27: Matriz de confusión

		Predicciones											
		0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	20933	615	2152	0	0	0	0	0	0	0	0	0
	1	4797	13514	0	0	0	0	0	0	0	93	0	0
	2	0	5196	0	1054	0	0	0	768	1036	38	46	0
	3	0	0	0	18	788	22555	1822	0	0	0	0	0
	4	1084	0	0	0	0	0	0	0	0	0	0	0
	5	11	692	0	1816	1931	0	3166	0	0	0	0	1472
	6	354	3219	0	0	17965	0	0	0	0	0	0	10752
	7	0	228	19335	10400	0	937	7735	13526	12804	16219	230	0
	8	0	1	106	0	0	0	4879	0	0	4053	42	0
	9	0	0	0	7495	0	0	2383	810	0	960	0	0
	10	0	0	0	600	0	0	68	771	1052	1512	23175	0
	11	0	0	0	0	0	0	0	0	0	0	0	0

Tabla 28: tabla de clases con precisión y cobertura

Clase	Precisión	Cobertura
0	0,482182756	0,883248945
1	0,375451464	0,734296892
2	0	0
3	8,00E-04	7,15E-04
4	0	0
5	0	0
6	0	0
7	0,375659612	0,166138502
8	0	0
9	4,10E-02	0,082417582
10	0,507611434	0,852711752
11	-	-

Tasa de acierto total: **29,1762 %**

➤ **Sujeto 2:**

Procedimiento básico

Tabla 29: Matriz de confusión

		Predicciones											
		0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	21463	0	0	0	0	0	0	0	0	0	0	0
	1	13	1301	0	0	0	0	0	0	2	0	0	0
	2	319	4	0	0	0	0	0	0	0	0	0	0
	3	767	0	0	24796	0	0	9421	2792	2049	0	0	0
	4	0	0	0	0	3296	1718	0	313	910	0	0	8099
	5	0	0	0	6308	0	16705	18626	2543	227	0	0	0
	6	0	0	0	0	5479	6172	0	875	700	0	0	4702
	7	0	0	0	0	0	0	0	2903	1979	0	0	0
	8	0	0	0	0	0	0	0	72	945	0	0	0
	9	710	19472	23007	0	0	0	0	6679	7510	20456	3331	0
	10	0	1545	2409	396	0	0	404	1157	823	209	25476	0
	11	0	0	0	0	0	0	0	0	45	0	0	0

Tabla 30: tabla de clases con precisión y cobertura

Clase	Precisión	Cobertura
0	0,44722037	1
1	0,046748114	0,988601824
2	0	0
3	4,83E-01	6,23E-01
4	0,110511316	0,229910714
5	0,386385715	0,37616249
6	0	0
7	0,09863414	0,594633347
8	0,034339911	0,92920354
9	4,35E-01	0,252029816
10	0,489875973	0,785835467
11	0	0

Tasa de acierto total: **45,283 %**

➤ **Sujeto 3:**

Procedimiento básico

Tabla 31: Matriz de confusión

		Predicciones											
		0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	20453	1786	0	0	0	0	0	0	46	218	0	0
	1	696	19917	15554	0	0	0	0	0	7	3362	1025	0
	2	0	6825	1268	13	0	0	0	294	1629	3063	18385	0
	3	0	210	671	2361	0	0	0	2622	3971	881	0	0
	4	0	0	0	0	0	0	0	1474	2455	33	0	0
	5	423	0	0	1190	0	0	0	477	214	31	0	0
	6	0	0	0	24729	0	0	0	398	197	114	0	0
	7	0	0	2895	0	0	0	0	2372	4139	0	0	0
	8	0	0	18	0	0	0	0	22	888	0	0	0
	9	0	0	2	0	0	0	0	2605	990	9458	986	0
	10	0	0	0	0	0	0	0	87	732	3128	7543	0
	11	469	0	0	0	0	0	0	15	0	0	0	0

Tabla 32: tabla de clases con precisión y cobertura

Clase	Precisión	Cobertura
0	0,466984794	0,908901035
1	0,465426588	0,491038189
2	0,052518224	0,040283382
3	9,36E-02	2,20E-01
4	0	0
5	0	0
6	0	0
7	0,093892253	0,25217946
8	0,037367447	0,956896552
9	2,93E-01	0,673598747
10	0,247970019	0,656483899
11	0	0

Tasa de acierto total: **37,0714 %**

➤ Sujeto 4:

Procedimiento básico

Tabla 33: Matriz de confusión

		Predicciones											
		0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	21279	0	0	0	0	0	0	102	300	0	0	0
	1	548	0	5	39	0	0	0	600	276	499	0	0
	2	708	24748	24149	16220	0	0	0	8391	11266	954	608	0
	3	0	0	0	0	0	0	0	0	0	4165	0	0
	4	0	0	0	210	1	6754	1952	0	0	0	0	0
	5	0	0	0	1970	0	15796	24738	0	0	0	0	0
	6	69	156	1	11123	0	0	0	7408	1058	11256	0	0
	7	0	0	0	0	0	0	0	0	0	660	0	0
	8	0	0	0	0	0	0	0	0	0	0	0	0
	9	0	0	246	0	0	0	0	0	0	1637	10758	0
	10	425	574	250	468	0	0	0	182	1379	822	13576	0
	11	0	0	0	0	0	59	0	0	0	0	0	0

Tabla 34: tabla de clases con precisión y cobertura

Clase	Precisión	Cobertura
0	0,244131617	0,98145842
1	0	0
2	0,268226853	0,277434401
3	0,00E+00	0,00E+00
4	1,52E-05	1,12E-04
5	0,193251609	0,371635611
6	0	0
7	0	0
8	-	-
9	2,42E-02	0,129499248
10	0,17085541	0,768047069
11	0	0

Tasa de acierto total: **33,4689 %**

➤ **Sujeto 5:**

Procedimiento básico

Tabla 35: Matriz de confusión

		Predicciones											
		0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	20733	105	0	0	0	0	0	0	1	1300	0	0
	1	61	1187	1384	0	0	159	144	127	987	2908	0	0
	2	819	1050	17881	0	85	78	0	1909	1256	12538	3400	0
	3	0	0	0	4167	88	1	0	3515	3188	0	0	7
	4	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0	0	0	0	0	0
	7	0	0	0	6006	20410	13454	8453	4033	408	0	0	7441
	8	0	0	0	7869	491	1338	8336	2864	3661	42	60	0
	9	2014	565	1422	12511	3227	7774	8999	287	0	18	1615	0
	10	43	23813	1384	0	0	1557	0	1515	3214	7590	27819	0
	11	0	0	0	0	0	0	0	0	0	0	0	0

Tabla 36: tabla de clases con precisión y cobertura

Clase	Precisión	Cobertura
0	0,850584615	0,936492163
1	0,245806585	0,17061952
2	0,830785671	0,458299159
3	5,34E-01	3,80E-01
4	-	-
5	-	-
6	-	-
7	0,525472313	0,066987792
8	0,501300835	0,148453023
9	4,92E-03	4,68E-04
10	0,884237628	0,415612161
11	-	-

Tasa de acierto total: **29,5194 %**

➤ **Sujeto 6:**

Procedimiento básico

Tabla 37: Matriz de confusión

		Predicciones											
		0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	10706	0	0	0	0	0	0	0	7	0	0	0
	1	9374	10632	447	0	0	0	0	0	53	39	20	0
	2	2374	10469	6904	957	0	14	19	31	0	175	3786	0
	3	0	0	0	17922	0	1878	17836	2019	337	0	0	0
	4	0	0	0	0	22062	698	0	0	0	0	0	253
	5	0	0	0	0	67	12967	0	0	0	0	0	0
	6	0	0	0	1405	0	2702	3776	883	3908	2938	0	0
	7	0	0	0	0	0	0	0	3513	261	0	0	0
	8	0	0	0	9	0	0	47	2921	4131	0	0	0
	9	215	100	0	329	0	920	2183	3369	2561	14779	11254	0
	10	661	1607	16713	4734	0	1249	2629	549	0	3098	22524	0
	11	0	0	0	0	62	4	0	0	0	0	0	0

Tabla 38: tabla de clases con precisión y cobertura

Clase	Precisión	Cobertura
0	0,424740141	0,999346588
1	0,423046315	0,516994894
2	0,322556531	0,27918638
3	5,53E-01	4,48E-01
4	6,02E-01	9,59E-01
5	4,72E-01	9,95E-01
6	2,07E-01	2,42E-01
7	0,195025815	0,930842607
8	0,221727229	0,58117614
9	5,05E-01	4,14E-01
10	0,608362143	0,418942043
11	0,00E+00	0,00E+00

Tasa de acierto total: **52,3685 %**

➤ **Sujeto 7:**

Procedimiento básico

Tabla 39: Matriz de confusión

		Predicciones											
		0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	24441	12216	13178	133	0	0	104	41	58	0	0	0
	1	72	0	12520	0	0	0	0	982	914	11758	21108	0
	2	0	0	0	2	0	0	0	1743	3668	7653	0	0
	3	0	0	0	21963	1284	20396	14332	3903	1387	2099	0	0
	4	675	0	0	0	0	0	0	2	0	0	0	0
	5	422	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	2	0	2184	12110	527	5327	0	0	0
	7	0	0	0	0	0	0	0	2721	47	0	0	0
	8	0	0	0	0	2211	0	0	6240	100	0	0	0
	9	0	0	14	3404	0	0	1862	1	0	0	7841	0
	10	0	0	0	6747	0	3	0	1462	111	0	409	0
	11	0	0	0	0	0	0	0	0	0	0	0	0

Tabla 40: tabla de clases con precisión y cobertura

Clase	Precisión	Cobertura
0	0,415112606	0,487153934
1	0	0
2	0	0
3	3,89E-01	3,36E-01
4	0,00E+00	0,00E+00
5	0,00E+00	0,00E+00
6	2,60E-01	6,01E-01
7	0,073227838	0,983020231
8	0,002895445	0,011694539
9	0,00E+00	0,00E+00
10	0,011737359	0,046839212
11	-	-

Tasa de acierto total: **26,8012 %**

➤ Sujeto 8:

Procedimiento básico

Tabla 41: Matriz de confusión

		Predicciones											
		0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	499	0	0	0	69	20	693	0	0	0	14543	19
	1	47	1297	0	0	304	290	6847	0	0	0	6	43
	2	9	12074	4297	0	0	0	0	0	0	29	285	0
	3	498	30	8521	11629	0	8405	0	2662	2478	4011	5614	0
	4	0	0	0	0	15821	7527	20412	0	0	0	0	8405
	5	0	118	0	2330	0	8988	380	3573	577	0	0	0
	6	0	0	0	6209	0	0	53	1388	3369	1364	106	0
	7	0	0	0	10175	0	0	0	4006	3227	0	0	0
	8	0	0	0	0	0	0	0	0	0	665	646	0
	9	267	0	12187	0	0	0	0	32	0	17354	11666	0
	10	22839	9379	64	0	0	0	0	18	0	797	19	0
	11	0	0	0	0	0	0	35	0	0	0	0	0

Tabla 42: tabla de clases con precisión y cobertura

Clase	Precisión	Cobertura
0	0,019682865	0,03149656
1	0,04959847	0,146819108
2	0,147409949	0,257397867
3	3,19E-01	2,65E-01
4	3,89E-01	3,03E-01
5	2,66E-01	5,63E-01
6	2,13E-03	4,24E-03
7	0,138812849	0,230124081
8	0	0
9	4,11E-01	4,18E-01
10	7,64E-04	5,74E-04
11	0,00E+00	0,00E+00

Tasa de acierto total: **24,6756%**

➤ **Todos los sujetos:**

Tabla 43: tasa de acierto de los sujetos y media por procedimiento básico

Sujeto	Tasa de acierto (%)
1	29,18
2	45,28
3	37,07
4	33,47
5	29,52
6	52,37
7	26,80
8	24,68
Media	34,80

Tabla 44: tasa de acierto de los sujetos y media por procedimiento con probabilidades de pertenencia

Sujeto	Tasa de acierto (%)
1	30,56
2	8,55
3	32,62
4	26,15
5	30,13
6	9,40
7	25,89
8	22,73
Media	23,25

Tasa de acierto de mejor sujeto → **52,37 %**

4.3.2 Experimentos dependientes del sujeto

En otros contextos, puede ser útil el reconocimiento de actividades realizadas por un sujeto específico al que se ha adaptado el clasificador. Para ver el rendimiento del sistema en este aspecto, se ha realizado una validación cruzada intra-sujeto, o sea, se dividen los datos del sujeto en diez partes iguales y, una vez por cada parte, se utiliza dicha parte como conjunto de test y el resto, como conjunto de entrenamiento.

Los resultados obtenidos se pueden analizar en una matriz de confusión y las métricas correspondientes a la mejor ejecución de cada sujeto, así como la media total de las mismas métricas contando con todas las ejecuciones para cada sujeto.

➤ Sujeto 1:

Procedimiento básico

Tabla 45: tasa de acierto de las ejecuciones y media

Ejecución (<i>fold</i>)	Tasa de acierto (%)
0	98,0178
1	97,1723
2	98,2646
3	97,1359
4	96,3511
5	97,6780
6	97,5000
7	97,8722
8	97,8236
9	95,2483
Media	97,31

Tabla 46: precisión y cobertura del mejor *fold* por clases

Clase	Precisión	Cobertura
0	0,869228285	0,882044561
1	0,830685619	1
2	0,847400151	1
3	0,838966203	1
4	0,834288052	1
5	0,85235144	0,990258365
6	0,834288052	1
7	0,796072508	0,986275733
8	0,783074451	0,988505747
9	0,844708589	1
10	0,854630294	0,997068677
11	0,749226006	1

Tasa de acierto de mejor *fold* → **98,2646 %**

Tabla 47: matriz de confusión del mejor *fold*

	Predicciones											
	0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	2692	360	0	0	0	0	0	0	0	0	0
	1	0	1987	0	0	0	0	0	0	0	0	0
	2	0	0	2249	0	0	0	0	0	0	0	0
	3	0	0	0	2110	0	0	0	0	0	0	0
	4	0	0	0	0	2039	0	0	0	0	0	0
	5	0	0	0	0	0	2338	23	0	0	0	0
	6	0	0	0	0	0	0	2039	0	0	0	0
	7	0	0	0	0	0	0	0	1581	22	0	0
	8	0	0	0	0	0	0	0	17	1462	0	0
	9	0	0	0	0	0	0	0	0	0	2203	0
	10	0	0	0	0	0	0	0	0	0	7	2381
	11	0	0	0	0	0	0	0	0	0	0	1210

Procedimiento con probabilidades de pertenencia

Tabla 48: tasa de acierto de las ejecuciones y media

Ejecución (<i>fold</i>)	Tasa de acierto (%)
0	97,7063
1	98,4709
2	98,3899
3	97,7710
4	97,0833
5	98,6245
6	97,8155
7	97,7912
8	98,8794
9	95,7659
Media	97,83

Tabla 49: precisión y cobertura del mejor *fold* por clases

Clase	Precisión	Cobertura
0	0,948889672	1
1	0,941767068	0,969408847
2	0,935181046	1
3	0,936514886	1
4	0,934359439	1
5	0,942	0,963190184
6	0,929951691	0,997926387
7	0,916570771	0,951044776
8	0,909600998	0,988482385
9	0,940695297	0,99913119
10	0,93993372	0,996486605
11	0,89298893	1

Tasa de acierto de mejor *fold* → **98,8794 %**

Tabla 50: matriz de confusión del mejor *fold*

	Predicciones											
	0	1	2	3	4	5	6	7	8	9	10	11
0	2692	0	0	0	0	0	0	0	0	0	0	0
1	72	2345	1	0	1	0	0	0	0	0	0	0
2	0	0	2092	0	0	0	0	0	0	0	0	0
3	0	0	0	2139	0	0	0	0	0	0	0	0
4	0	0	0	0	2064	0	0	0	0	0	0	0
5	0	0	0	0	24	2355	66	0	0	0	0	0
6	0	0	0	0	0	0	1925	1	0	3	0	0
7	0	0	0	0	0	0	0	1593	77	5	0	0
8	0	0	0	0	0	0	0	17	1459	0	0	0
9	0	0	0	0	0	0	0	0	2	2300	0	0
10	0	0	0	0	0	0	0	0	0	8	2269	0
11	0	0	0	0	0	0	0	0	0	0	0	1210

➤ Sujeto 2:

Procedimiento básico

Tabla 51: tasa de acierto de las ejecuciones y media

Ejecución (<i>fold</i>)	Tasa de acierto (%)
0	95,2338
1	95,9594
2	94,3616
3	96,3993
4	96,1716
5	97,1055
6	95,3959
7	91,4286
8	94,5700
9	95,5015
Media	95,21

Tabla 52: precisión y cobertura del mejor *fold* por clases

Clase	Precisión	Cobertura
0	0,935590421	0,956118143
1	0,93235039	0,994909764
2	0,939976914	0,967908082
3	0,95223515	0,999357326
4	0,844	1
5	0,938727416	0,972335232
6	0,947100712	0,973509934
7	0,910755149	0,934820904
8	0,896207585	1
9	0,930512249	0,930512249
10	0,947208122	0,952040816
11	0,895652174	1

Tasa de acierto de mejor *fold* → **97,1055 %**

Tabla 53: matriz de confusión del mejor *fold*

	Predicciones											
	0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	2266	0	0	0	0	0	19	84	0	1	0
	1	0	2150	0	0	0	0	7	3	0	1	0
	2	0	8	2443	0	0	0	0	10	0	63	0
	3	0	0	0	3110	0	0	2	0	0	0	0
	4	0	0	0	0	844	0	0	0	0	0	0
	5	0	0	0	0	17	2390	51	0	0	0	0
	6	0	0	0	0	0	75	2793	0	1	0	0
	7	0	0	0	7	0	0	0	1592	97	1	6
	8	0	0	0	0	0	0	0	0	1347	0	0
	9	0	15	0	0	0	0	133	0	2089	8	0
	10	13	77	23	0	0	0	2	26	0	2799	0
	11	0	0	0	0	0	0	0	0	0	0	1339

Procedimiento con probabilidades de pertenencia

Tabla 54: tasa de acierto de las ejecuciones y media

Ejecución (<i>fold</i>)	Tasa de acierto (%)
0	95,1566
1	95,9516
2	93,8291
3	96,7235
4	94,7090
5	93,1267
6	93,8985
7	89,4836
8	94,6742
9	95,0385
Media	94,26

Tabla 55: precisión y cobertura del mejor *fold* por clases

Clase	Precisión	Cobertura
0	0,888703627	0,998594848
1	0,887103594	0,992901088
2	0,904438082	0,942910448
3	0,921677911	1
4	0,770223752	0,995550612
5	0,90191036	0,995539335
6	0,915318744	0,994486561
7	0,857371795	0,919243986
8	0,833125	0,844740177
9	0,873818526	0,97521097
10	0,914313222	0,930133856
11	0,828736369	1

Tasa de acierto de mejor *fold* → **96,7235 %**

Tabla 56: matriz de confusión del mejor *fold*

	Predicciones											
	0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	2132	0	0	0	0	0	0	0	3	0	0
	1	0	2098	12	0	0	0	0	0	0	3	0
	2	4	59	2527	0	0	0	0	15	20	55	0
	3	0	0	0	3142	0	0	0	0	0	0	0
	4	0	0	0	0	895	3	0	0	0	0	1
	5	0	0	0	0	11	2455	0	0	0	0	0
	6	0	0	0	0	0	16	2886	0	0	0	0
	7	0	0	0	0	0	0	0	1605	82	59	0
	8	75	0	0	0	0	0	0	0	1333	170	0
	9	0	0	0	0	0	0	0	45	2	1849	0
	10	65	0	0	0	0	0	0	71	74	4	2849
	11	0	0	0	0	0	0	0	0	0	0	1292

➤ Sujeto 3:

Procedimiento básico

Tabla 57: tasa de acierto de las ejecuciones y media

Ejecución (<i>fold</i>)	Tasa de acierto (%)
0	98,9154
1	97,8020
2	97,6404
3	97,6981
4	98,7250
5	97,8135
6	97,8135
7	97,4674
8	98,1193
9	98,6558
Media	98,07

Tabla 58: precisión y cobertura del mejor *fold* por clases

Clase	Precisión	Cobertura
0	0,974533879	1
1	0,981182796	1
2	0,972139303	0,997447677
3	0,980926431	1
4	-	-
5	-	-
6	-	-
7	0,944111776	0,944111776
8	0,963206307	0,931385006
9	0,97357244	1
10	0,980212014	0,993197279
11	-	-

Tasa de acierto de mejor *fold* → **98,9154 %**

Tabla 59: matriz de confusión del mejor *fold*

	Predicciones											
	0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	2143	0	0	0	0	0	0	0	0	0	0
	1	0	2920	0	0	0	0	0	0	0	0	0
	2	0	5	1954	0	0	0	0	0	0	0	0
	3	0	0	0	2880	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	946	56	0	0
	8	0	0	0	0	0	0	0	108	1466	0	0
	9	0	0	0	0	0	0	0	0	2063	0	0
	10	0	0	19	0	0	0	0	0	0	2774	0
	11	0	0	0	0	0	0	0	0	0	0	0

Procedimiento con probabilidades de pertenencia

Tabla 60: tasa de acierto de las ejecuciones y media

Ejecución (<i>fold</i>)	Tasa de acierto (%)
0	98,1596
1	95,3732
2	85,4159
3	97,3866
4	98,7250
5	97,4616
6	93,9425
7	95,1424
8	85,7736
9	91,5488
Media	93,89

Tabla 61: precisión y cobertura del mejor *fold* por clases

Clase	Precisión	Cobertura
0	0,96792371	1
1	0,974544204	0,996833216
2	0,96318408	0,996910402
3	0,975201072	1
4	-	-
5	-	-
6	-	-
7	0,929924242	0,937917861
8	0,951379763	0,966622163
9	0,963293651	1
10	0,974509128	0,968835616
11	-	-

Tasa de acierto de mejor *fold* → **98,725 %**

Tabla 62: matriz de confusión del mejor *fold*

	Predicciones											
	0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	2233	0	0	0	0	0	0	0	0	0	0
	1	0	2833	9	0	0	0	0	0	0	0	0
	2	0	0	1936	0	0	0	0	0	0	6	0
	3	0	0	0	2910	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	982	65	0	0
	8	0	0	0	1	0	0	0	49	1448	0	0
	9	0	0	0	0	0	0	0	0	1942	0	0
	10	0	0	91	0	0	0	0	0	0	2829	0
	11	0	0	0	0	0	0	0	0	0	0	0

➤ Sujeto 4:

Procedimiento básico

Tabla 63: tasa de acierto de las ejecuciones y media

Ejecución (<i>fold</i>)	Tasa de acierto (%)
0	95,3542
1	95,7089
2	94,5222
3	93,0247
4	95,5337
5	93,0598
6	95,2622
7	94,8813
8	95,1134
9	95,7711
Media	94,82

Tabla 64: precisión y cobertura del mejor *fold* por clases

Clase	Precisión	Cobertura
0	0,972759227	1
1	0,975513428	0,95109742
2	0,975298805	0,936495792
3	0,977982955	0,998187749
4	-	-
5	0,973413379	0,997802198
6	0,977205882	0,925165332
7	0,958583834	0,970906631
8	0,956059532	0,868641339
9	0,967708333	0,965696466
10	0,975030205	0,947183099
11	-	-

Tasa de acierto de mejor *fold* → **95,7711 %**

Tabla 65: matriz de confusión del mejor *fold*

	Predicciones											
	0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	2214	0	0	0	0	0	0	0	0	0	0
	1	14	2470	0	4	0	0	0	99	10	0	0
	2	1	8	2448	0	0	0	0	3	1	80	73
	3	0	0	0	2754	0	0	0	2	3	0	0
	4	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	2270	5	0	0	0	0
	6	0	0	0	214	0	1	2658	0	0	0	0
	7	13	0	0	0	0	0	0	1435	30	0	0
	8	64	58	0	0	0	0	0	82	1349	0	0
	9	10	20	0	0	0	0	0	1	8	1858	27
	10	0	0	0	0	0	0	0	17	34	84	2421
	11	0	0	0	0	0	0	0	0	0	0	0

Procedimiento con probabilidades de pertenencia

Tabla 66: tasa de acierto de las ejecuciones y media

Ejecución (<i>fold</i>)	Tasa de acierto (%)
0	88,3702
1	86,7413
2	94,5222
3	93,0466
4	95,5337
5	90,9274
6	91,5798
7	80,9834
8	93,2086
9	93,1882
Media	90,81

Tabla 67: precisión y cobertura del mejor *fold* por clases

Clase	Precisión	Cobertura
0	0,937198068	1
1	0,94552381	0,929588015
2	0,94591528	0,938813814
3	0,950052393	0,999265246
4	-	-
5	0,938494624	1
6	0,949092204	0,915521978
7	0,917197452	0,899488927
8	0,897048236	0,922962963
9	0,927814235	0,976101965
10	0,945168712	0,962890625
11	-	-

Tasa de acierto de mejor *fold* → **95,5337 %**

Tabla 68: matriz de confusión del mejor *fold*

	Predicciones											
	0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	2134	0	0	0	0	0	0	0	0	0	0
	1	55	2482	0	7	0	0	0	105	21	0	0
	2	0	1	2501	0	0	0	0	0	0	82	80
	3	0	0	0	2720	0	0	0	0	2	0	0
	4	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	2182	0	0	0	0	0
	6	0	0	0	246	0	0	2666	0	0	0	0
	7	50	0	0	1	0	0	0	1584	126	0	0
	8	38	52	0	0	0	0	3	9	1246	2	0
	9	9	4	0	0	0	0	0	13	4	1838	15
	10	0	11	0	0	0	0	0	9	12	63	2465
	11	0	0	0	0	0	0	0	0	0	0	0

➤ Sujeto 5:

Procedimiento básico

Tabla 69: tasa de acierto de las ejecuciones y media

Ejecución (<i>fold</i>)	Tasa de acierto (%)
0	91,0512
1	90,3011
2	91,1551
3	91,0252
4	96,2162
5	90,5536
6	90,3568
7	91,5190
8	92,1651
9	91,0032
Media	91,53

Tabla 70: precisión y cobertura del mejor *fold* por clases

Clase	Precisión	Cobertura
0	0,916961826	0,995726496
1	0,927166034	0,942786943
2	0,90378477	0,987543597
3	0,932845321	0,940629012
4	0,918702151	0,989399293
5	0,913982878	0,982471516
6	0,923963964	0,991492653
7	0,81888412	0,954954955
8	0,855281207	0,725421757
9	0,921091997	0,998783455
10	0,939245609	0,995422643
11	0,776008493	0,98384926

Tasa de acierto de mejor *fold* → **96,2162 %**

Tabla 71: matriz de confusión del mejor *fold*

	Predicciones											
	0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	2330	0	0	0	0	0	0	0	10	0	0
	1	0	2686	163	0	0	0	0	0	0	0	0
	2	0	0	1982	0	0	0	0	0	0	25	0
	3	0	0	0	2931	0	184	0	0	1	0	0
	4	0	0	0	0	2520	0	0	0	0	0	27
	5	0	0	0	39	0	2242	0	1	0	0	0
	6	0	0	0	22	0	0	2564	0	0	0	0
	7	0	0	0	0	0	0	0	954	45	0	0
	8	0	0	0	1	0	0	0	471	1247	0	0
	9	0	0	0	0	0	0	0	0	2463	3	0
	10	0	0	15	0	0	0	0	0	0	3262	0
	11	0	0	0	0	12	0	0	0	0	0	731

Procedimiento con probabilidades de pertenencia

Tabla 72: tasa de acierto de las ejecuciones y media

Ejecución (<i>fold</i>)	Tasa de acierto (%)
0	79,8967
1	81,0330
2	87,2860
3	93,0860
4	96,2162
5	92,5624
6	87,3305
7	92,8483
8	89,7850
9	90,3126
Media	89,04

Tabla 73: precisión y cobertura del mejor *fold* por clases

Clase	Precisión	Cobertura
0	0,916961826	0,995726496
1	0,927166034	0,942786943
2	0,90378477	0,987543597
3	0,932845321	0,940629012
4	0,918702151	0,989399293
5	0,913982878	0,982471516
6	0,923963964	0,991492653
7	0,81888412	0,954954955
8	0,855281207	0,725421757
9	0,921091997	0,998783455
10	0,939245609	0,995422643
11	0,776008493	0,98384926

Tasa de acierto de mejor *fold* → **96,2162 %**

Tabla 74: matriz de confusión del mejor *fold*

	Predicciones											
	0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	2330	0	0	0	0	0	0	0	10	0	0
	1	0	2686	163	0	0	0	0	0	0	0	0
	2	0	0	1982	0	0	0	0	0	0	25	0
	3	0	0	0	2931	0	184	0	0	1	0	0
	4	0	0	0	0	2520	0	0	0	0	0	27
	5	0	0	0	39	0	2242	0	1	0	0	0
	6	0	0	0	22	0	0	2564	0	0	0	0
	7	0	0	0	0	0	0	0	954	45	0	0
	8	0	0	0	1	0	0	0	471	1247	0	0
	9	0	0	0	0	0	0	0	0	2463	3	0
	10	0	0	15	0	0	0	0	0	0	3262	0
	11	0	0	0	0	12	0	0	0	0	0	731

➤ **Sujeto 6:**

Procedimiento básico

Tabla 75: tasa de acierto de las ejecuciones y media

Ejecución (<i>fold</i>)	Tasa de acierto (%)
0	96,8518
1	97,8192
2	95,5981
3	96,3842
4	92,9982
5	93,3327
6	96,9163
7	96,3600
8	94,9774
9	95,6022
Media	95,68

Tabla 76: precisión y cobertura del mejor *fold* por clases

Clase	Precisión	Cobertura
0	0,964809384	0,984608807
1	0,964056483	0,953045685
2	0,965058236	0,968280467
3	0,968072976	1
4	0,961308153	0,985363551
5	0,961538462	0,944669366
6	0,968796434	1
7	0,938098747	0,996867659
8	0,928143713	0,961879433
9	0,960170697	0,99704579
10	0,9776	0,970611597
11	-	-

Tasa de acierto de mejor *fold* → **97,8192 %**

Tabla 77: matriz de confusión del mejor *fold*

	Predicciones											
	0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	2303	0	0	12	0	0	3	4	17	0	0
	1	36	2253	75	0	0	0	0	0	0	0	0
	2	0	0	2320	0	0	0	0	0	4	0	72
	3	0	0	0	2547	0	0	0	0	0	0	0
	4	0	0	0	0	2087	0	0	0	0	0	31
	5	0	0	0	0	123	2100	0	0	0	0	0
	6	0	0	0	0	0	0	2608	0	0	0	0
	7	0	0	0	0	0	0	0	1273	4	0	0
	8	22	8	0	0	0	0	0	13	1085	0	0
	9	0	0	0	0	0	0	0	1	0	2025	5
	10	0	0	32	0	0	0	0	7	0	72	3666
	11	0	0	0	0	0	0	0	0	0	0	0

Procedimiento con probabilidades de pertenencia

Tabla 78: tasa de acierto de las ejecuciones y media

Ejecución (<i>fold</i>)	Tasa de acierto (%)
0	97,5612
1	97,5532
2	92,6636
3	93,6713
4	72,4201
5	88,8785
6	97,2105
7	96,9848
8	76,6365
9	77,7894
Media	89,14

Tabla 79: precisión y cobertura del mejor *fold* por clases

Clase	Precisión	Cobertura
0	0,969723183	0,995559503
1	0,97037664	0,95861204
2	0,970526316	0,97339527
3	0,973524962	0,991525424
4	0,967830882	0,991525424
5	0,967089798	0,946182153
6	0,973604827	1
7	0,951048951	0,997067449
8	0,940778342	0,962770563
9	0,965483235	0,997452878
10	0,980998914	0,941636269
11	-	-

Tasa de acierto de mejor *fold* → **97,5612 %**

Tabla 80: matriz de confusión del mejor *fold*

	Predicciones											
	0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	2242	0	0	0	0	0	0	10	0	0	0
	1	36	2293	61	0	0	0	0	0	2	0	0
	2	0	5	2305	0	0	0	0	0	0	0	58
	3	0	0	0	2574	0	0	10	0	12	0	0
	4	0	0	0	0	2106	0	0	0	0	0	18
	5	0	0	0	0	117	2057	0	0	0	0	0
	6	0	0	0	0	0	0	2582	0	0	0	0
	7	0	0	0	0	0	0	0	1360	4	0	0
	8	27	2	0	0	0	0	0	14	1112	0	0
	9	0	0	0	0	0	0	0	0	0	1958	5
	10	0	0	39	0	0	0	0	21	0	164	3614
	11	0	0	0	0	0	0	0	0	0	0	0

➤ Sujeto 7:

Procedimiento básico

Tabla 81: tasa de acierto de las ejecuciones y media

Ejecución (<i>fold</i>)	Tasa de acierto (%)
0	90,8885
1	92,6726
2	94,7866
3	92,5120
4	92,1864
5	92,8549
6	94,1572
7	93,9835
8	93,1458
9	92,0413
Media	92,92

Tabla 82: precisión y cobertura del mejor *fold* por clases

Clase	Precisión	Cobertura
0	0,936473165	0,996503497
1	0,878151261	0,999203187
2	0,938537619	0,950966356
3	0,94530022	0,944409548
4	0,635983264	0,713615023
5	0,91769158	0,93179635
6	0,937410072	0,915349491
7	0,892791128	0,989078498
8	0,869760479	0,902173913
9	0,922459893	0,927835052
10	0,941922563	0,976132826
11	-	-

Tasa de acierto de mejor *fold* → **94,7866 %**

Tabla 83: matriz de confusión del mejor *fold*

	Predicciones											
	0	1	2	3	4	5	6	7	8	9	10	11
0	2565	0	0	0	0	0	9	0	0	0	0	0
1	0	1254	0	0	0	0	0	1	0	0	0	0
2	0	0	2657	68	1	0	5	0	0	0	63	0
3	0	0	0	3007	0	93	84	0	0	0	0	0
4	0	0	0	98	304	4	20	0	0	0	0	0
5	0	0	0	42	10	1940	90	0	0	0	0	0
6	0	0	0	0	0	240	2606	1	0	0	0	0
7	0	0	0	0	0	0	0	1449	11	3	2	0
8	0	0	0	9	4	1	3	102	1162	0	7	0
9	0	0	0	0	0	0	0	161	0	2070	0	0
10	0	0	0	0	0	0	0	28	41	0	2822	0
11	0	0	0	0	0	0	0	0	0	0	0	0

Procedimiento con probabilidades de pertenencia

Tabla 84: tasa de acierto de las ejecuciones y media

Ejecución (<i>fold</i>)	Tasa de acierto (%)
0	82,4369
1	87,9064
2	91,4007
3	89,2520
4	88,7572
5	89,4213
6	80,9654
7	86,7257
8	90,3546
9	87,3502
Media	87,46

Tabla 85: precisión y cobertura del mejor *fold* por clases

Clase	Precisión	Cobertura
0	0,98767809	0,998442974
1	0,97301855	1
2	0,984983576	0,970411466
3	0,990015601	0,93213866
4	0,896774194	0,985815603
5	0,982251803	0,974147415
6	0,988342441	0,717724868
7	0,980148883	0,806533946
8	0,974004874	0,926584235
9	0,982646421	0,967948718
10	0,988338192	0,989059081
11	-	-

Tasa de acierto de mejor *fold* → **91,4007 %**

Tabla 86: matriz de confusión del mejor *fold*

	Predicciones											
	0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	2565	0	0	0	0	0	0	0	0	4	0
	1	0	1154	0	0	0	0	0	0	0	0	0
	2	0	0	2099	0	1	0	0	0	0	63	0
	3	0	29	0	3173	12	93	97	0	0	0	0
	4	0	0	0	0	278	4	0	0	0	0	0
	5	0	0	0	42	0	1771	5	0	0	0	0
	6	0	71	558	0	28	409	2713	1	0	0	0
	7	0	0	0	0	0	0	0	1580	10	261	108
	8	0	0	0	9	0	1	2	76	1199	0	7
	9	0	0	0	0	0	0	0	60	0	1812	0
	10	0	0	0	0	0	0	0	25	5	0	2712
	11	0	0	0	0	0	0	0	0	0	0	0

➤ Sujeto 8:

Procedimiento básico

Tabla 87: tasa de acierto de las ejecuciones y media

Ejecución (<i>fold</i>)	Tasa de acierto (%)
0	94,8805
1	93,2371
2	93,7733
3	90,1354
4	96,6475
5	92,8667
6	90,6600
7	91,8984
8	93,9199
9	90,7583
Media	92,88

Tabla 88: precisión y cobertura del mejor *fold* por clases

Clase	Precisión	Cobertura
0	0,953468697	1
1	0,95329087	0,965176268
2	0,959213941	0,960282108
3	0,964945825	0,984395319
4	0,902411022	0,888637648
5	0,957594449	0,997189884
6	0,962546816	0,964188267
7	0,908180301	0,885993485
8	0,885773624	0,87308086
9	0,955982393	1
10	0,966473636	1
11	0,834337349	0,90228013

Tasa de acierto de mejor *fold* → **96,6475 %**

Tabla 89: matriz de confusión del mejor *fold*

	Predicciones											
	0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	2254	0	0	0	0	0	0	0	0	0	0
	1	77	2245	0	0	0	0	2	0	0	2	0
	2	0	38	2587	0	0	0	13	0	0	56	0
	3	48	0	0	3028	0	0	0	0	0	0	0
	4	0	0	0	0	1572	0	0	0	0	0	197
	5	0	0	0	0	5	2484	2	0	0	0	0
	6	0	0	0	0	1	32	2827	0	0	0	72
	7	31	0	1	0	0	0	1088	108	0	0	0
	8	0	0	0	0	0	0	124	853	0	0	0
	9	0	0	0	0	0	0	0	0	2389	0	0
	10	0	0	0	0	0	0	0	0	0	3171	0
	11	0	0	0	0	60	0	0	0	0	0	554

Procedimiento con probabilidades de pertenencia

Tabla 90: tasa de acierto de las ejecuciones y media

Ejecución (<i>fold</i>)	Tasa de acierto (%)
0	92,2803
1	92,6237
2	93,1136
3	90,1354
4	82,9173
5	93,0095
6	90,6600
7	91,8984
8	89,7149
9	90,7583
Media	90,71

Tabla 91: precisión y cobertura del mejor *fold* por clases

Clase	Precisión	Cobertura
0	0,875804619	1
1	0,865408289	0,989676208
2	0,882940757	0,909558824
3	0,893747975	1
4	0,690958904	0,853179973
5	0,875136116	0,998757249
6	0,889135255	0,907533139
7	0,763176895	0,727460427
8	0,659751037	0,675876727
9	0,881759193	0,998367347
10	0,90876217	0,990300091
11	0,645021645	0,685057471

Tasa de acierto de mejor *fold* → **93,1136 %**

Tabla 92: matriz de confusión del mejor *fold*

	Predicciones											
	0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	2313	0	0	0	0	0	0	0	0	0	0
	1	2	2109	15	0	0	0	0	0	5	0	0
	2	38	110	2474	0	0	0	0	29	0	6	62
	3	0	0	0	2759	0	0	0	0	0	0	0
	4	0	0	0	0	1261	1	1	0	0	0	215
	5	0	0	0	0	0	2411	0	0	0	0	3
	6	0	0	0	0	174	84	2807	0	0	0	28
	7	52	3	0	23	0	0	0	1057	309	9	0
	8	47	0	0	174	0	0	0	84	636	0	0
	9	0	0	0	0	0	0	0	1	0	2446	3
	10	0	0	0	0	0	0	0	4	28	0	3267
	11	0	0	0	0	236	16	22	0	0	0	596

➤ Todos los sujetos:

Tabla 93: tasa de acierto media de cada sujeto y media total por procedimiento básico

Sujeto	Tasa de acierto (%)
1	97,31
2	95,21
3	98,07
4	94,82
5	91,53
6	95,68
7	92,92
8	92,88
Media	94,80

Tabla 94: tasa de acierto media de cada sujeto y media total por procedimiento con probabilidades de pertenencia

Sujeto	Tasa de acierto (%)
1	97,83
2	94,26
3	93,89
4	90,81
5	89,04
6	89,14
7	87,46
8	90,71
Media	91,64

Tasa de acierto media de mejor sujeto → **98,07 %**

4.3.3 Experimentos parcialmente dependientes del sujeto

Otro enfoque analizado ha sido el reconocimiento de actividades realizadas por varios sujetos, cuando el clasificador ha aprendido de otro conjunto distinto de datos de actividades realizadas por los mismos sujetos. Para ello, la validación cruzada se ha efectuado a partir de un compendio de todos los datos de todos los sujetos, dividiendo el mismo en diez partes iguales, nuevamente, y utilizando cada una como conjunto de test con las respectivas partes restantes como conjunto de entrenamiento.

A semejanza de lo visto en los experimentos anteriores, los resultados se presentan en una matriz de confusión junto con las medidas de rendimiento, en este caso en referencia a la mejor prueba. A su vez, se dispone la media total de la tasa de acierto para las diez pruebas.

➤ Con 11 atributos:

Procedimiento básico

Tabla 95: tasa de acierto de las ejecuciones y media

Ejecución (<i>fold</i>)	Tasa de acierto (%)
0	46,1833
1	52,8975
2	70,2267
3	50,4939
4	46,7233
5	41,7651
6	52,2949
7	59,9543
8	52,9733
9	53,8074
Media	52,73

Tabla 96: precisión y cobertura del mejor *fold* por clases

Clase	Precisión	Cobertura
0	0,449486954	0,988077592
1	0,514943301	0,766082546
2	0,496999893	0,787454376
3	0,494413182	0,711082714
4	0,140948808	0,576858266
5	0,335574269	0,424093369
6	0,207846446	0,381671702
7	0,30247762	0,52383403
8	0,025986825	0,817292007
9	0,437329578	0,781358745
10	0,504485962	0,843131202
11	0,115580256	1

Tasa de acierto de mejor *fold* → **70,2267%**

Tabla 97: matriz de confusión del mejor *fold*

	Predicciones											
	0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	15332	106	0	0	0	6	0	0	68	0	5
	1	4309	19935	346	22	14	5	10	29	314	367	671
	2	62	301	18554	829	7	20	246	693	579	1408	863
	3	3	0	0	18363	0	1294	1579	1483	3101	1	0
	4	0	134	0	0	3081	500	0	0	0	0	1626
	5	0	0	0	1785	1407	9484	8640	273	221	0	553
	6	0	57	0	352	2358	3268	4927	729	665	22	41
	7	0	0	0	1473	0	0	0	8143	5929	0	0
	8	0	0	0	0	0	0	0	501	111	1	0
	9	213	432	137	114	4	226	101	731	538	14595	1588
	10	140	3	166	313	11	262	468	572	594	1028	19118
	11	0	0	0	0	0	0	0	0	0	0	2454

Procedimiento con probabilidades de pertenencia

Tabla 98: tasa de acierto de las ejecuciones y media

Ejecución (<i>fold</i>)	Tasa de acierto (%)
0	42,6038
1	53,4662
2	64,1762
3	46,9906
4	43,6153
5	36,6258
6	53,4323
7	57,9747
8	48,6924
9	45,1059
Media	49,27

Tabla 99: precisión y cobertura del mejor *fold* por clases

Clase	Precisión	Cobertura
0	0,505376699	0,732466415
1	0,570712854	0,747805537
2	0,5419276	0,728601939
3	0,528266272	0,748440007
4	0,127335157	0,507892293
5	0,358968878	0,471132806
6	0,173965736	0,361616856
7	0,270067663	0,575220321
8	0,02140573	0,817955112
9	0,464043177	0,782391226
10	0,577665118	0,517393608
11	-	-

Tasa de acierto de mejor *fold* → **64,1762 %**

Tabla 100: matriz de confusión del mejor *fold*

Etiquetas	Predicciones											
	0	1	2	3	4	5	6	7	8	9	10	11
0	15321	154	0	3	2033	636	0	0	0	249	0	2521
1	4320	19935	338	32	14	372	201	69	317	406	654	0
2	62	301	17740	1270	7	20	397	964	832	2004	751	0
3	3	0	0	16792	0	1160	835	1294	2351	1	0	0
4	0	111	0	0	2188	385	0	0	0	0	0	1624
5	0	0	0	330	196	8397	8075	171	105	0	0	549
6	0	23	0	289	976	2825	3158	613	479	22	73	275
7	0	0	0	463	0	0	0	5548	3634	0	0	0
8	0	0	0	0	0	0	0	0	328	73	0	0
9	213	432	959	113	4	226	101	731	538	12983	294	0
10	140	12	166	3959	1464	1044	3204	3263	3858	1862	20510	159
11	0	0	0	0	0	0	0	0	0	0	0	0

➤ Con 22 atributos:

Procedimiento básico

Tabla 101: tasa de acierto de las ejecuciones y media

Ejecución (<i>fold</i>)	Tasa de acierto (%)
0	42,4038
1	47,0820
2	39,3720
3	56,0437
4	47,3781
5	61,3235
6	46,7593
7	54,2677
8	55,7215
9	67,3025
Media	51,77

Tabla 102: precisión y cobertura del mejor *fold* por clases

Clase	Precisión	Cobertura
0	0,721756454	0,847383296
1	0,606364864	0,868252901
2	0,61911255	0,510469855
3	0,740968278	0,580610978
4	0,565171074	0,869552642
5	0,546167132	0,789362519
6	0,712222453	0,732593383
7	0,344736585	0,61445993
8	0,216273089	0,573110285
9	0,555349207	0,622111598
10	0,769336636	0,606027754
11	0,080384088	0,497031383

Tasa de acierto de mejor *fold* → **67,3025 %**

Tabla 103: matriz de confusión del mejor *fold*

		Predicciones											
		0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	17390	2517	311	3	0	0	69	0	29	203	0	0
	1	155	10327	761	0	0	0	0	157	32	462	0	0
	2	20	2805	10897	147	7	1555	211	535	221	2221	2728	0
	3	84	8	19	19177	423	4412	1976	2569	2414	1836	111	0
	4	0	0	0	0	9019	0	0	0	0	0	0	1353
	5	7	0	0	468	790	8222	343	352	143	0	0	91
	6	1	0	0	451	1591	509	17161	1016	1227	495	5	969
	7	0	0	0	0	0	134	289	3527	1497	293	0	0
	8	3	0	0	26	4	87	154	1104	1850	0	0	0
	9	66	260	0	231	45	2621	318	1057	488	8373	0	0
	10	589	166	6812	2066	47	0	200	478	734	3444	22360	0
	11	0	0	0	0	235	128	230	0	0	0	0	586

Procedimiento con probabilidades de pertenencia

Tabla 104: tasa de acierto de las ejecuciones y media

Ejecución (<i>fold</i>)	Tasa de acierto (%)
0	41,4847
1	45,7870
2	37,3015
3	55,0531
4	46,0293
5	48,9681
6	45,1750
7	51,2516
8	50,9973
9	63,4895
Media	48,55

Tabla 105: precisión y cobertura del mejor *fold* por clases

Clase	Precisión	Cobertura
0	0,627921593	0,854584336
1	0,521592	0,868691117
2	0,541972921	0,336981359
3	0,639546389	0,601201975
4	0,433173527	0,855542725
5	0,553911992	0,808308444
6	0,591065877	0,726777673
7	0,257039768	0,625262355
8	0,14373531	0,527363184
9	0,409770688	0,575126815
10	0,699368394	0,604327793
11	0,05826208	0,553875236

Tasa de acierto de mejor *fold* → **63,4895 %**

Tabla 106: matriz de confusión del mejor *fold*

		Predicciones											
		0	1	2	3	4	5	6	7	8	9	10	11
Etiquetas	0	15985	2459	0	0	6	9	5	0	29	212	0	0
	1	129	10327	761	0	0	144	0	78	96	353	0	0
	2	1190	2814	11208	2660	1491	1368	987	1833	1800	5071	2728	110
	3	321	7	19	16806	897	930	4409	1885	1576	993	111	0
	4	0	0	0	0	7409	0	0	0	0	0	0	1251
	5	43	55	14	468	828	11908	603	350	135	0	237	91
	6	5	0	0	388	1256	359	13880	844	976	424	5	961
	7	0	0	0	0	0	134	289	3277	1248	293	0	0
	8	2	0	0	26	4	87	153	1065	1590	0	88	0
	9	52	255	0	200	3	2611	294	985	458	6576	0	0
	10	588	166	6798	2021	44	0	200	478	727	3405	22035	0
	11	0	0	0	0	223	118	131	0	0	0	0	586

4.4 Resumen y análisis

Con el objetivo de una posterior extracción de conclusiones, a continuación se expone un resumen de los resultados obtenidos con cada tipo de experimento, así como un breve análisis de los mismos. Se hace saber que, al haberse observado la consecución en general de un mayor acierto mediante el procedimiento básico, en este resumen y análisis se omiten los resultados referentes al procedimiento con probabilidades de pertenencia.

4.4.1 Experimentos independientes del sujeto

En los resultados de los experimentos de clasificación independiente del sujeto con **11 atributos**, se han observado los siguientes aspectos:

- El mejor resultado se ha obtenido para el sujeto 2, con un 48,5671% de acierto; mientras que el peor resultado obtenido ha sido para el

sujeto 6, con un 27,5463%. La media de acierto para todos los sujetos es del 37,13%. Esto puede deberse a que el sujeto 2 es de los sujetos con menos instancias aún conteniendo todas las clases del protocolo, de manera que se dispone de un conjunto de entrenamiento amplio, puesto que no se retiran demasiadas instancias del conjunto total, manteniéndose un conjunto de test representativo del mismo. Sin embargo, el sujeto 6 sí comprende un número notablemente mayor de instancias, perdiéndose información relevante en el conjunto de entrenamiento.

- Las clases para las que se ha logrado un mayor acierto son 0, 1 y 10 (estar tumbado, estar sentado y planchar); frente a las clases 6 y 8 (marcha nórdica y bajar escaleras), cuya predicción ha sido la menos acertada.
- Es destacable la frecuencia de confusión de la predicción entre dos grupos principales de clases: el compuesto por las clases 0, 1, 2 y 10 (estar tumbado, estar sentado, estar de pie y planchar); y el formado por las clases 3, 4, 5, 6, 7 y 8 (caminar, correr, ir en bici, marcha nórdica, subir escaleras y bajar escaleras). La clase 9 (pasar la aspiradora) es confundida con menos frecuencia con clases de ambos grupos, y la clase 11 (saltar a la comba) apenas es confundida con ninguna otra clase.

En los resultados de los experimentos de clasificación independiente del sujeto con **22 atributos**, destaca lo siguiente:

- El mejor resultado se ha obtenido para el sujeto 6, con un 52,37% de acierto; mientras que el peor resultado obtenido ha sido para el sujeto 8, con un 24,68%. La media de acierto para todos los sujetos es del 34,8%. Esto puede deberse a que el sujeto 8 comprende un número muy elevado de instancias, perdiéndose información relevante en el conjunto de entrenamiento, mientras que, en el caso del sujeto 6, el aprendizaje ha sido favorecido con el uso de una configuración más amplia de atributos.
- Las clases para las que se ha logrado un mayor acierto son 0 y 10 (estar tumbado y planchar); frente a las clases 4, 5, 6 y 11 (correr, ir en bici, marcha nórdica y saltar a la comba), cuya predicción ha sido la menos acertada.
- Es destacable la frecuencia de confusión de la predicción entre dos grupos principales de clases: el compuesto por las clases 0, 1, 2, 9 y 10 (estar tumbado, estar sentado, estar de pie, pasar la aspiradora y planchar); y el formado por las clases 3, 4, 5, 6, 7 y 8 (caminar, correr, ir en bici, marcha nórdica, subir escaleras y bajar escaleras).

La clase 11 (saltar a la comba) apenas es confundida con ninguna otra clase.

4.4.2 Experimentos dependientes del sujeto

De los resultados de los experimentos de clasificación dependiente del sujeto, se han deducido las siguientes observaciones:

- El mejor resultado se ha obtenido para el sujeto 3, con un 98,07% de acierto; mientras que el peor resultado obtenido ha sido para el sujeto 5, con un 91,53%. La media de acierto para todos los sujetos es del 94,8%. Se entiende que el experimento del sujeto 3 ha logrado un mejor resultado porque contiene un número reducido de clases, por tanto, un conjunto más fácil de aprender para el algoritmo. EL sujeto 5, por el contrario, es uno de los sujetos con mayor número de instancias y mayor variedad de clases, complicando el proceso de aprendizaje.
- Las clases para las que se ha logrado un mayor acierto son la 0, 1 y 3 y 9 (estar tumbado, estar sentado, caminar y pasar la aspiradora); frente a las clases 6, 7 y 8 (marcha nórdica, subir escaleras y bajar escaleras), cuya predicción ha sido la menos acertada.

4.4.3 Experimentos parcialmente dependientes del sujeto

En los resultados de los experimentos de clasificación parcialmente dependiente del sujeto con **11 atributos**, se han observado los siguientes aspectos:

- El mejor resultado obtenido ha sido un 70,2267% de acierto; mientras que el peor resultado obtenido ha sido un 41,7651%. La media de acierto para todos los *folds* es del 52,73%.
- Las clases para las que se ha logrado un mayor acierto son 0, 10 y 11 (estar tumbado, planchar y saltar a la comba); frente a las clases 5, 6 y 7 (ir en bici, marcha nórdica y subir escaleras), cuya predicción ha sido la menos acertada.
- Es destacable la frecuencia de confusión de la predicción entre los siguientes grupos de clases: 1 y 0 (estar sentado y estar tumbado);

2, 9 y 10 (estar de pie, pasar la aspiradora y planchar); 3, 7 y 8 (caminar, subir escaleras y bajar escaleras); 5 y 6 (ir en bici y marcha nórdica); 4 y 11 (correr y saltar a la comba).

En los resultados de los experimentos de clasificación independiente del sujeto con **22 atributos**, se han observado los siguientes aspectos:

- El mejor resultado obtenido ha sido un 67,3025% de acierto; mientras que el peor resultado obtenido ha sido un 39,372%. La media de acierto para todos los *folds* es del 51,77%.
- Las clases para las que se ha logrado un mayor acierto son 0, 1 y 4 (estar tumbado, estar sentado y correr); frente a la clase 11 (saltar a la comba), cuya predicción ha sido la menos acertada.
- Es destacable la frecuencia de confusión de la predicción entre los siguientes grupos de clases: 0, 1, 2 y 10 (estar tumbado, estar sentado, estar de pie y planchar); 4, 5, 6 y 11 (correr, ir en bici, marcha nórdica y saltar a la comba).

5. Planificación

Para llevar a cabo este proyecto, se elaboró una planificación de las tareas implicadas. Sin embargo, durante la realización del mismo, dos imprevistos afectaron de manera significativa al cumplimiento de los plazos inicialmente determinados, por lo que fue necesaria la modificación de la planificación.

Se debe aclarar que las tareas relativas al Algoritmo Evolutivo, a excepción de la parte coevolutiva del mismo, no se incluyen en ninguna de las planificaciones puesto que fueron completadas con anterioridad a este proyecto, en los meses de noviembre y diciembre de 2014, para la práctica de la que se parte.

Los detalles de las distintas planificaciones, tareas involucradas y problemas que derivaron en cambios en la temporización de las mismas se explican en los siguientes apartados.

5.1 Tareas

El proceso planificado se compone de diferentes tareas, abarcadas por las tareas principales que se describen a continuación:

- Investigación y análisis del problema: incluye la búsqueda de información y bibliografía referente al estado del arte y los campos de investigación relacionados con el proyecto propuesto. Además, en esta fase se definen los objetivos y de éstos se extraen los requisitos.

- Diseño de las mejoras del sistema: comprende la identificación de alternativas de diseño viables y la selección de la más adecuada para cada una de las funcionalidades a incorporar al sistema del que se parte.

- Desarrollo de las mejoras del sistema: consiste en la implementación y depuración de las funcionalidades que se pretende añadir al sistema inicial. A su vez, consta de la realización de pruebas para la configuración de parámetros de los elementos que forman parte del sistema.
- Experimentación y evaluación del sistema: consta de los experimentos para la obtención de resultados y las conclusiones derivadas.
- Documentación del proyecto: engloba la organización de la información a incluir en el informe, la redacción de los apartados que lo conforman y la revisión y últimas correcciones del documento.

Para una mejor visualización y comprensión de la planificación de tareas, éstas se disponen tanto en un diagrama de Gantt, como en una tabla explicativa, con su secuenciación y temporización según la planificación correspondiente. Como ya se ha mencionado, se tuvo que considerar una planificación inicial y otra modificada debido a la sucesión de dos inconvenientes.

5.2 Planificación inicial

Para la estimación inicial de la duración del proyecto, se tuvo en cuenta la carga de 12 ECTS que tiene el Trabajo Fin de Grado que, siendo cada crédito equivalente a una cantidad de 25 a 30 horas de trabajo, implica la dedicación de una cantidad de 300 a 360 horas aproximadamente. De esta manera, las horas a invertir en cada tarea quedaron organizadas como se indica en la siguiente tabla y en su correspondiente diagrama de Gantt:

Tabla 107: Planificación inicial

ID	Nombre tarea	Duración (días)	Inicio	Fin
1	Investigación y análisis	25	16/02/2015	20/03/2015
1.1	Investigación del estado del arte	15	16/02/2015	06/03/2015
1.2	Definición de objetivos	5	09/03/2015	13/03/2015
1.3	Extracción de requisitos	5	16/03/2015	20/03/2015
2	Diseño y desarrollo	70	23/03/2015	26/06/2015
2.1	Diseño clustering-PSO	5	23/03/2015	27/03/2015
2.2	Desarrollo clustering-PSO	13	30/03/2015	15/04/2015
2.3	Depuración y parámetros clustering-PSO	15	16/04/2015	06/05/2015
2.4	Diseño coevolución	5	07/05/2015	13/05/2015
2.5	Desarrollo coevolución	12	14/05/2015	29/05/2015
2.6	Despuración y parámetros coevolución	20	01/06/2015	26/06/2015
3	Experimentación y evaluación	22	07/07/2015	05/08/2015
3.1	Experimentos	16	07/07/2015	28/07/2015
3.2	Análisis de resultados	6	29/07/2015	05/08/2015
4	Documentación del proyecto	17	06/08/2015	28/08/2015
4.1	Organización de la información	3	06/08/2015	10/08/2015
4.2	Redacción de apartados	11	11/08/2015	25/08/2015
4.3	Revisión del documento	3	26/08/2015	28/08/2015

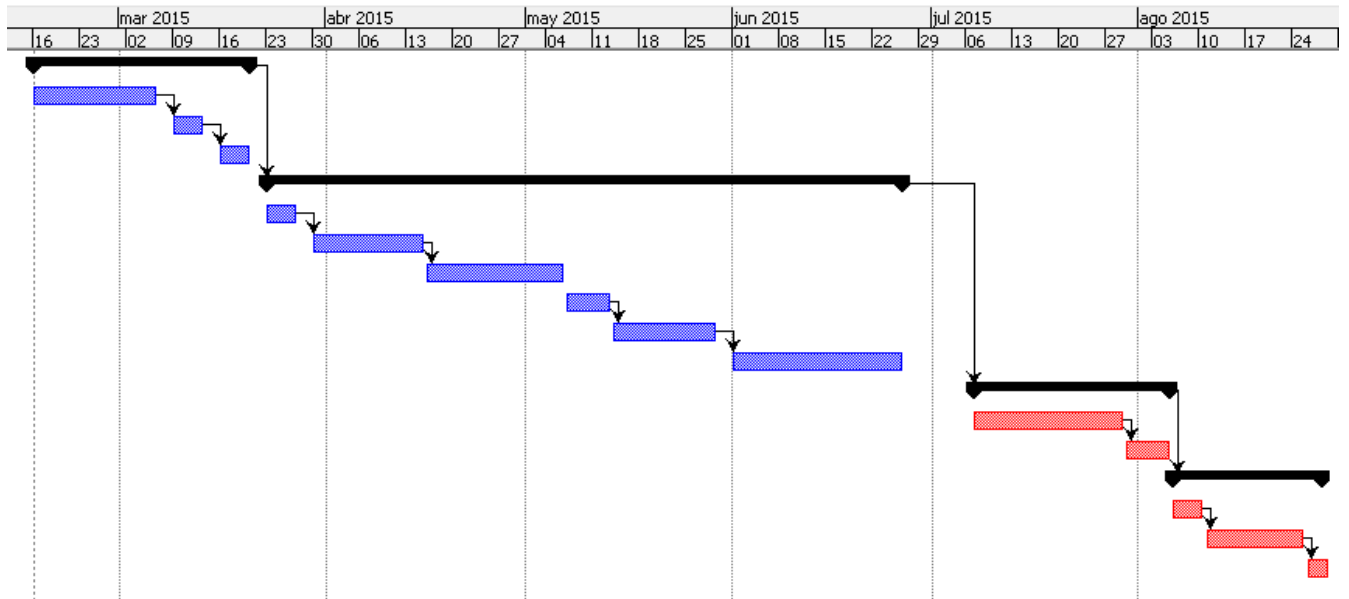


Ilustración 11: Planificación inicial

Teniendo en cuenta que la estimación de horas a dedicar se encuentra entre 300 y 360, y siendo laborables todos los días considerados para esta planificación inicial, es decir, 134, se calculó una jornada de unas 2,5 horas de media diaria, 335 horas en total.

5.3 Planificación final

Desafortunadamente, la planificación inicial tuvo que verse alterada principalmente por dos causas:

- Un problema personal de salud que mermó el avance del proyecto durante la última semana de febrero y el mes de abril.
- Un fallo importante en la selección de atributos, arrastrado hasta la primera semana de agosto y detectado durante la fase de experimentación, haciendo ineludible la realización de nuevas pruebas. La detección y solución de este problema se detalla en la sección [4.1.2 Preprocesamiento de datos](#) de este documento.

La planificación final, resultante de hacer los reajustes oportunos para solventar el primer inconveniente y, posteriormente, el segundo, se recoge en la tabla y el diagrama de Gantt que se muestran a continuación:

Tabla 108: Planificación final

ID	Nombre tarea	Duración (días)	Inicio	Fin
1	Investigación y análisis	30 (-5)	16/02/2015	27/03/2015
1.1	Investigación del estado del arte	20 (-5)	16/02/2015	13/03/2015
1.2	Definición de objetivos	5	16/03/2015	20/03/2015
1.2	Extracción de requisitos	5	23/03/2015	27/03/2015
2	Diseño y desarrollo	66 (-4)	27/04/2015	27/07/2015
2.1	Diseño clustering-PSO	4 (-1)	27/04/2015	30/04/2015
2.2	Desarrollo clustering-PSO	12 (-1)	01/05/2015	18/05/2015
2.3	Depuración y parámetros clustering-PSO	13 (-2)	19/05/2015	04/06/2015
2.4	Diseño coevolución	3 (-2)	05/06/2015	09/06/2015
2.5	Desarrollo coevolución	9 (-3)	10/05/2015	22/06/2015
2.6	Depuración y parámetros coevolución	25 (+5)	23/06/2015	27/07/2015
3	Experimentación y evaluación	32 (+10)	28/07/2015	09/09/2015
3.1	Experimentos	22 (+6)	28/07/2015	26/08/2015
3.2	Análisis de resultados	10 (+4)	27/08/2015	09/09/2015
4	Documentación del proyecto	20 (+3)	10/09/2015	08/10/2015
4.1	Organización de la información	2 (-1)	10/09/2015	14/09/2015
4.2	Redacción de apartados	12 (+1)	14/09/2015	30/09/2015
4.3	Revisión del documento	6 (+3)	30/09/2015	08/10/2015

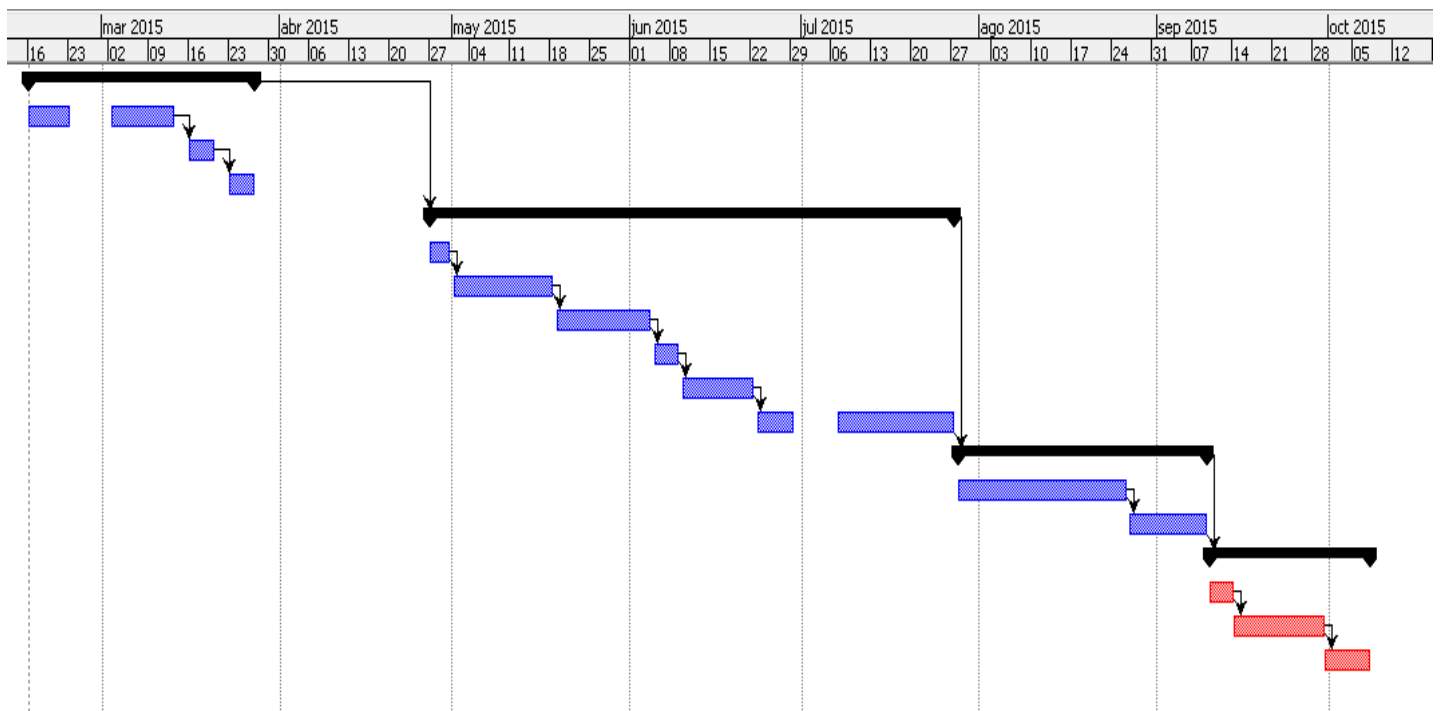


Ilustración 12: Planificación final

Al reajustar el proceso a la planificación modificada, que finalmente constaba de 148 días efectivos, se estimó una media de 2,5 horas diarias de trabajo, dedicándose unas 370 horas en total aproximadamente.

6. Marco regulador

Con intención de respetar el marco regulador, tanto técnico como legal, que pudiera afectar a la elaboración de este trabajo, se verificaron los aspectos de este tipo correspondientes a las herramientas y los datos empleados en el mismo.

Por un lado, las siguientes herramientas utilizadas se distribuyen bajo licencias aprobadas como software libre y de código abierto, por la Free Software Foundation (FSF) y la Open Source Initiative (OSI), respectivamente:

- Eclipse Luna Service Release 1 (4.4.1) → Eclipse Public License
- Netbeans IDE 7.0.1 → GNU General Public License (v2.0)
- ProjectLibre → Common Public Attribution License (v1.0)
- Weka 3.6 → GNU General Public License (v3.0)

Esto supone que estas herramientas: pueden ser libremente utilizadas, cambiadas y compartidas (modificadas o sin modificar) por cualquiera, según la Open Source Initiative (OSI) [231]; o que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software de las mismas, según la Free Software Definition (FSD) [232].

Sin embargo, también fue empleado software no libre, concretamente los sistemas operativos Windows 7 Professional y Windows Server 2008 Enterprise, adquiridos con las claves de producto pertinentes, a través del programa DreamSpark Premium ligado al programa MSDN Academic Alliance, gracias al vigente Contrato Campus entre la Universidad Carlos III de Madrid y Microsoft. A su vez, este acuerdo permite el uso de Microsoft Office en cualquier ordenador de la Universidad, herramienta que fue de utilidad para la documentación de este proyecto.

Por otro lado, los datos utilizados para la evaluación y entrenamiento del sistema, es decir, los recogidos en el conjunto PAMAP2 Physical Activity Monitoring, se encuentran libremente disponibles en el UCI Machine Learning Repository para la investigación académica. No hay restricciones, legales o de otro tipo, respecto al uso de estos datos con propósitos científicos [131] y los trabajos de origen fueron referenciados convenientemente en este documento (véase sección [4.1.1 Adquisición de datos](#)).

Sin embargo, el uso de datos para estos propósitos implica ciertas cuestiones éticas, más aún en el caso de datos personales, y supone una actuación responsable a la hora de aplicarse. Hay que tener en cuenta la dificultad de conseguir unos datos completamente anónimos sin retirar toda la información útil, por ejemplo, el 87% de la población estadounidense resulta ser identificable en registros públicos, a partir únicamente de su código postal, género y fecha de nacimiento [233]. Además, en los datos no debe contenerse características susceptibles de discriminación, tales como la raza, el sexo o la religión. También, es importante que resultados puramente estadísticos no se expongan como determinantes, y las conclusiones extraídas se respalden con otros argumentos o, por lo menos, incluyan la advertencia correspondiente. Al fin y al cabo, en qué medida afecten todos estos aspectos dependerá de la situación y la aplicación concreta de los datos.

Por consiguiente, para este proyecto se consideran las anteriores cuestiones mencionadas, manteniéndose el carácter anónimo de los datos, así como evitándose el uso y divulgación de información de carácter conflictivo, como el sexo, edad y otras características físicas o personales de los individuos participantes en el proceso de recolección de datos.

7. Presupuesto

Una vez definida la planificación final del proyecto, se puede especificar el presupuesto ligado a su cumplimiento. En este apartado, se detallan los costes estimados, desglosados en costes de personal, costes de software, costes de hardware y otros costes.

7.1 Costes de personal

Los costes de personal son los derivados del pago de sueldos a los integrantes del proyecto, en este caso, un ingeniero informático. En la tabla se detalla el cálculo ejecutado, teniendo en cuenta las horas invertidas en todo el proceso y considerando un sueldo medio de 13,5 euros por hora de trabajo.

Tabla 109: Costes de personal

Trabajador	Horas trabajadas	Sueldo / Hora	Sueldo total
Ingeniero informático	370	13,5 €/h	$370 \times 13.5 = 4.995 \text{ €}$

Por lo tanto, el coste total de personal se estima en 4.995 euros.

7.2 Costes de software

Con respecto a los costes vinculados al software empleado, todos los programas no gratuitos de por sí, se encontraban disponibles para su uso académico gratuito por vía de los recursos de la Universidad Carlos III de Madrid. Sin embargo, para que el trabajo resultara más eficiente, se decidió hacer un uso adicional de algunos de estos programas en un ordenador personal, debiendo incluirse los costes de las licencias pertinentes. A continuación, se incluye una tabla descriptiva con todos los costes de software.

Tabla 110: Costes de software

Software	Licencias necesarias	Coste / Licencia	Coste total
Microsoft Office 2007 (servidor)	1	0	0
Microsoft Office 2007 (personal)	1	0***	0
Windows 7 Home Premium (personal)	1	0**	0
Microsoft Office 2007 (universidad)	1	0	0
Windows 7 Enterprise	1	0	0
Windows Server 2008 Enterprise	1	0	0
Eclipse Luna 4.4.1	0*	0	0
Netbeans IDE 7.0.1	0*	0	0
ProjectLibre	0*	0	0

*Estos programas no necesitaron de una licencia al ser software libre y de código abierto.

**Este sistema operativo estaba incluido con el ordenador.

***Este programa estaba incluido con el sistema operativo.

Por consiguiente, el total de los costes de software se resume en 0 euros.

7.3 Costes de hardware

Para el cálculo de los costes de hardware se toma en consideración los equipos destinados a llevar a cabo el proyecto, en este caso, un servidor, un ordenador de sobremesa y un portátil. Tanto el servidor como el ordenador de sobremesa han sido proporcionados por la Universidad Carlos III de Madrid, por lo que no se incluyen en el cálculo. Para obtener el coste, se aplica la siguiente fórmula, en referencia al ordenador portátil personal empleado:

$$\text{coste HW} = \text{semanas de uso} * \text{coste} * \text{porcentaje de uso} / \text{depreciación}$$

El período de depreciación corresponde al tiempo que debe pasar para que el precio del ordenador descienda, y se establece en 52 semanas.

Tabla 111: Costes de hardware

Hardware	Semanas	Coste	Uso	Depreciación	Coste total
Ordenador portátil (personal)	29	400	70%	52 semanas	156 €

Determinándose un coste total de hardware de 156 euros.

7.4 Otros costes

Estos costes comprenden los gastos asociados al material fungible y los márgenes de error y beneficio, es decir, los gastos indirectos.

Para la realización de este proyecto, se consideran unos gastos aproximados de 30 € en gastos de impresión y se añade un 15%, aplicable a los gastos directos ya descritos, para cubrir los gastos ligados a la disponibilidad de conexión a internet, el uso de electricidad, etc.

Además, debe tenerse en cuenta un margen de error que asegure la finalización del proyecto, el cual se establece en un 10%, así como un margen de beneficio que garantice su rentabilidad, determinándose la adición de un 20% en este concepto.

7.5 Coste total

La suma de los costes de personal, software, hardware y otros gastos asciende a la cantidad de nueve mil trescientos setenta y dos euros con diecisiete céntimos.

Tabla 112: Coste total

Coste total	
Personal	4.995,00 €
Software	0,00 €
Hardware	156,00 €
Otros costes	30,00 €
Gastos indirectos (15%)	777,15 €
Total	5.958,15 €
Margen de error	595,82 €
Beneficio	1.191,63 €
Total sin IVA	7.745,60 €
IVA 21%	1.626,57 €
Total Presupuesto	9.372,17 €

8. Conclusiones

8.1 Cumplimiento de objetivos

En este apartado, se analiza el cumplimiento de los objetivos definidos al inicio de la realización de este trabajo, principalmente enfocados a la investigación y la apertura de frentes de estudio alternativos, en concreto para el reconocimiento de actividades. Dichos objetivos y el análisis respectivo de su consecución se presentan a continuación:

- Se han implementado y empleado técnicas de computación evolutiva, tanto un algoritmo evolutivo para la clasificación de actividades, como un enjambre de partículas para la optimización del clustering previo de los datos.
- Se ha llevado a cabo la integración de la coevolución, a ambas partes del sistema, consiguiendo un uso eficiente de los datos disponibles durante la ejecución de la fase de aprendizaje.
- Se ha utilizado un conjunto de datos público, concretamente el PAMAP2 [28], ya propuesto en otros trabajos, que contiene datos recogidos mediante sensores *wearable*.
- La evaluación del sistema implementado se ha realizado mediante tres enfoques distintos, referentes al nivel de especialización o generalización del clasificador en relación a los sujetos.
- Los resultados obtenidos, aún pudiendo ser mejorables por investigadores expertos, son comparables al estado del arte, en el caso de la clasificación dependiente del sujeto, no así en la clasificación independiente o parcialmente dependiente del sujeto. Durante la fase de aprendizaje, se pudo observar que el algoritmo evolutivo presentaba buena convergencia para los tres tipos de clasificación, llegando a tasas de acierto similares a las alcanzadas para los experimentos de evaluación dependientes del sujeto. De esto puede deducirse que el sistema tiende a la especialización de los clasificadores, posiblemente solucionable mediante una mejor elaboración de los conjuntos de entrenamiento.

- Se ha llevado a cabo un estudio del estado del arte abarcando tanto el reconocimiento de actividades, como la computación evolutiva y biológica y el aprendizaje automático, concretamente, la clasificación y el clustering.

Por otro lado, en el aspecto académico personal, este trabajo ha supuesto:

- La posibilidad de adquirir y aplicar nuevos conocimientos, ampliando la experiencia con respecto a lo aprendido durante la realización del grado, en las áreas de reconocimiento de actividades, computación evolutiva y aprendizaje automático.
- La oportunidad de completar un proyecto en su totalidad, suponiendo el uso de las herramientas adecuadas y la aplicación práctica de los conocimientos teóricos, con la toma de decisiones que conlleva, además de la elaboración de una documentación convenientemente desarrollada y precisa.

8.2 Líneas futuras de trabajo

En esta sección se sugieren diversas posibilidades y variaciones del sistema objeto del trabajo realizado, a investigar en posibles futuros estudios con el fin de mejorar el mismo. De esta manera, se exponen los siguientes puntos a tener en cuenta:

- Preprocesamiento de datos que conserve un mayor número de instancias. Para ello, podría aplicarse una sustitución de los atributos desconocidos en las instancias pertinentes (el 1% del total), similar a la realizada en el atributo de la frecuencia cardíaca al representar un porcentaje notablemente más alto.
- Selección de atributos alternativa que favorezca la generalización de los clasificadores.
- Aplicación de la fase de segmentación de manera complementaria al clustering.
- Estratificación de los conjuntos de entrenamiento cuando sea posible.

- Método de poda de árboles de decisión más eficaz en la reducción del *overfitting*, mediante, por ejemplo, la eliminación de las ramas que sólo produzcan predicciones erróneas.
- Consideración de distintos pesos para los atributos en la fase de clustering, en base a alguna magnitud como la entropía o la ganancia de información de cada uno.
- Combinación de los diversos módulos del sistema con algoritmos alternativos externos.

En conclusión, todos los objetivos propuestos al inicio de este proyecto han sido alcanzados, abriéndose nuevas líneas de investigación en el campo del reconocimiento de actividades, a pesar de no lograr unos resultados muy elevados en dos de los tres enfoques estudiados. Es una muestra más de lo abierta que está dicha área de investigación, así como de la gran variedad de aplicaciones que puede tener la computación evolutiva, aún más amplia si se combina con otras disciplinas como, en este caso, el aprendizaje automático.

ANEXO I. Competencias en inglés

Introduction

Nowadays, there are plenty of fields of applications in which human activity automatic recognition is very useful, or even necessary in some cases. Specially, in health and wellness, this discipline is used extensively, but also in other different areas such as ubiquitous computing, entertainment, personal daily activities logging, or sportive and professional monitoring.

In developed countries as well as in developing countries, human population's lifestyle is getting more sedentary every day, and research about the links between physical inactivity and a sort of diseases, from non transmissible diseases, like heart disease, hypertension and diabetes, to psychological issues like depression.

The World Health Organization (WHO) [11] declares the following facts on this problem, reflecting the seriousness of this population's propensity, the reaction about it and its relation to some health affections:

- Insufficient physical activity is 1 of the 10 leading risk factors for death worldwide.
- Insufficient physical activity is a key risk factor for noncommunicable diseases (NCDs) such as cardiovascular diseases, cancer and diabetes.
- Physical activity has significant health benefits and contributes to prevent NCDs.
- Globally, 1 in 4 adults is not active enough.
- More than 80% of the world's adolescent population is insufficiently physically active.

- Policies to address insufficient physical activity are operational in 56% of WHO Member States.
- WHO Member States have agreed to reduce insufficient physical activity by 10% by 2025.

Due to this tendency, activity recognition systems are presented as a good solution for the need of automatic methods to quantify routines and human activity patterns, being applied in the research for initiatives that promote physical activity [57, 136], the improvement of diagnosis and treatment of neurological, degenerative and respiratory disorders, such as Parkinson disease [137], multiple sclerosis [139] or chronic lung diseases [140].

Moreover, recent development in sensors miniaturization make possible the recompilation of many features of human motion unobtrusively, bringing new horizons, including the use of wearable sensors, for the application of activity recognition in ubiquitous computing, among others.

However, all those data full of potentially useful information have no practical sense unless they are interpreted by algorithms capable of extracting the knowledge, in this case, referring the human activities involved. The development of activity recognition systems faces multiple challenges, becoming an attractive field of work, not to mention again its diverse applications.

Last but not least, despite that the recognition techniques' state-of-the-art has numerous approaches of very diverse nature, it is a relatively new field, and the approaches that include evolutionary computing techniques, although they seem to be promising, they aren't abundant yet. On the contrary, these methods have already been extensively applied to other branches of bioinformatics, so it might be interesting to follow that not so explored line of research.

Objectives

The objectives or goals of this paper are more linked to research ones, so that new study areas are opened with an alternative approach for activity recognition, than to the development of a practical or potentially commercial application. Thus the aims proposed for this work are the following:

- The implementation and application of evolutionary computing techniques. From this point, other three more specific points are derived:
 - Design and usage of an evolutionary algorithm for activities classification.
 - Development of a particle swarm to optimize a previous clustering task.
 - Integration of coevolution, to both parts above, with the aim to make the most of the available data for the training stage.
- The usage of a public data set, already proposed in other works, and, in particular, which contains data obtained with wearable sensors.
- Evaluation of the implemented system, based on three different approaches, according to whether the classifier is adapted or not to specific users.
- To obtain performance results that, even though they might be outperformed by expert researchers, can be compared to the state-of-the-art, and are promising enough to promote future studies in the matter.
- The study of the state-of-the-art that covers from activity recognition, to evolutionary computing and machine learning.

In addition, in a more personal academic level, this work pursues some extra goals:

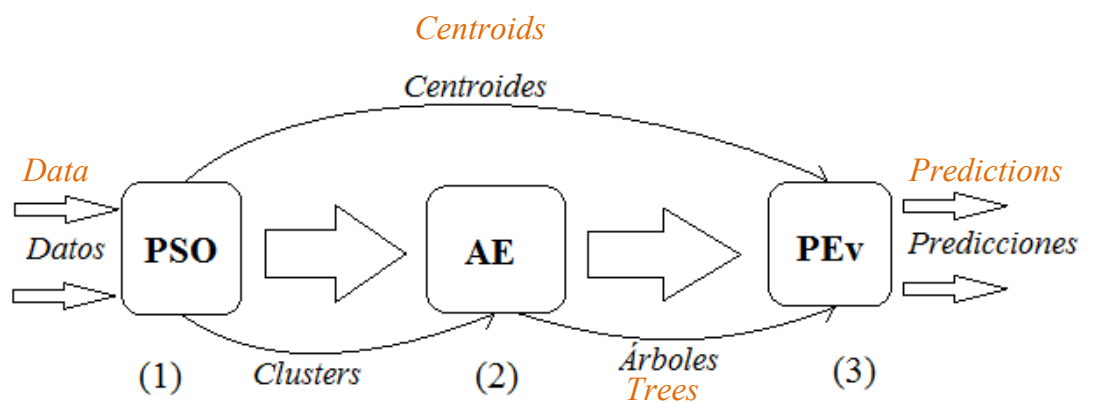
- Acquisition of new knowledge in the areas of activity recognition, evolutionary computing and machine learning.
- Extension of personal experience in the usage of evolutionary computing techniques, as well as classification and clustering methods, in addition to the human activity recognition standard methodology.
- Development of a whole project, including making the pertinent decisions, using the correct tools and the practical application of theoretical knowledge, in addition to a precise and complete documentation of it.

Design of the solution

For the system design, the decision of dividing it in three modules was taken:

- An evolutionary algorithm (AE) to train and generate the activities classifiers. Concretely, the representation chosen for the classifiers were decision trees.
- A particle swarm (PSO) to cluster the examples according to their affinity with each other, through the optimization of the centroids configuration.
- An evaluating program (PEv) that puts together the products of the former modules for the simulation of the system's final classification, in order to apply it to new data.

Therefore, once the system is implemented, the mechanism, that might be observed in the figure, works as follows: (1) the execution of the clustering module, with the particle swarm which produces the different clusters and their respective centroids; (2) afterwards, the learning phase with the evolutionary algorithm, building the classifier tree for each cluster of examples; and (3) the introduction of the centroids values and the trees of every cluster in the program to apply to every test instance the decision tree corresponding to the cluster that was assigned to the instance regarding how close to the centroids it was.



General procedure of the modules

Module 1: PSO

In module number one, where the procedure begins, the particle swarm groups the input data by minimizing the Euclidean distance of every example to the centroids of the cluster where it belongs. The cluster membership of an instance is decided by calculating the distance of the instance attributes to each cluster's centroids, choosing the closest one. In this paper, one more factor is added to the generic formula used to modify the particles' velocity: the mean of the instances in each cluster. This way, a combination between the normal particle swarm optimization algorithm and clustering algorithm k-means is applied.

Module 2: AE

The second module is where the learning phase of decision trees takes part. For this aim, an evolutionary algorithm comprised with a fusion between genetic programming, well suited for decision trees representation, and evolutionary strategies, very useful to adapt the numeric values to be compared with the instances' attributes at the nodes. In this part of the system, the evolutionary algorithm is executed with the instances in each cluster as the training set, one at a time, building a different decision tree adapted to each cluster.

Also, to solve the “ideal trainer” problem, in both modules, competitive coevolution is involved, incorporating another population filled with example sets that force the classifiers population to improve, at the same time as classifiers force the sets to try and make it more difficult for them.

In addition, when the decision trees are built, a pruning method is applied to them, in order to get rid of useless rules and branches that may cause noise.

Module 3: PEv

And to conclude this section, module number three, which is the program destined to evaluate the final model generated, gets the centroids produced in the clustering phase (1) and the decision trees built in the learning phase (2). Through the centroids, the program distributes the unknown or test instances in the available clusters and, once the trees are parsed, they predict the activities of the instances in their linked cluster. Finally, the program evaluates and shows the results obtained, based on the predefined metrics.

Experiments and results

In order to comply with the utilization of common benchmarked datasets, in this research the PAMAP2 dataset is used [28], which was introduced and publicly made available by A. Reiss and D. Stricker for this purpose.

The PAMAP2 portion data used here was acquired through 3 wearable sensors, worn by 8 individuals while they performed a fixed protocol of 12 activities, such as lying, standing, walking, running or cycling, among others. From the original data, different configurations of attributes were selected, using Weka toolkit, for the different experiments:

- **Subject dependent classification:**
 - *Subject 1*: 9 attributes.
 - *Subject 2*: 8 attributes.
 - *Subject 3*: 16 attributes .
 - *Subject 4*: 16 attributes.
 - *Subject 5*: 14 attributes.
 - *Subject 6*: 13 attributes.
 - *Subject 7*: 11 attributes.
 - *Subject 8*: 16 attributes.
- **Subject independent and partially dependent classification:**
 - *Selected at least for 4 subjects*: 11 attributes.
 - *Selected at least for 3 subjects*: 22 attributes.

Basically, the methodology followed matches the activity recognition chain (ARC), except of data segmentation stage, which is meant to be replaced by the clustering phase.

In the end, three approaches, already mentioned above, were chosen for the test or evaluation of the developed classifiers, and the results obtained for each one of them are shown below:

➤ **Subject independent classification:**

The following aspects were observed in the results of the experiments carried for 11 attributes:

- The best result was obtained for subject 2, with an accuracy rate of 48,5671%; whereas the worst result was obtained for subject 6, with an accuracy rate of 27,5463%. The mean value of all the subjects' accuracy rate is 37,13%. This could be due to subject 2 being one of the subjects with the smallest amount of examples while containing every class from the protocol, leaving a wide training set and keeping a representative test set. However, subject 6 has a bigger number of examples, leaving less relevant information for the training set.
- The activities that got the best accuracy rate are 0, 1 and 10 (lying, sitting and ironing); versus classes 6 and 8 (nordic walk and descending stairs), which got the worst prediction.
- There is a noticeably frequent confusion between two main groups of activities: the one including activities 0, 1, 2 and 10 (lying, sitting, standing and ironing); and the one involving activities 3, 4, 5, 6, 7 and 8 (normal walk, running, cycling, nordic walk, ascending stairs and descending stairs).

Regarding the experiments carried with 22 attributes, these are the remarkable aspects deduced from the results:

- The best result was found for subject 6, with an accuracy rate of 52,37%, being the worst accuracy rate for subject 8, with 24,68%. The accuracy mean for all the subjects is 34,8%.
- The activities with best prediction rate are 0 and 10 (lying and ironing), and the worst prediction rate was obtained for the activities 4, 5, 6 and 11 (running, cycling, nordic walk and rope jumping).
- The most frequent confusion in prediction happens between the activities that belong to two main groups: activities 0, 1, 2, 9 and 10 (lying, sitting, standing, vacuuming and ironing); and activities 3, 4, 5, 6, 7 and 8 (normal walk, running, cycling, nordic walk, ascending stairs and descending stairs).

➤ **Subject dependent classification:**

From the subject dependent experiments the next observations were made:

- The best accuracy rate was achieved for subject 3, with 98,07%; whereas the worst accuracy rate was 91,53%, obtained for subject 5. The mean value for all the subjects is an accuracy rate of 94,8%. It is understood that the experiment with subject 3 got the best result due to subject 3 containing a reduced number of activities, therefore having a training set that is easier to learn for the algorithm. On the contrary, subject 5 is one of the subjects with the biggest amount of examples and wider variety of activities, turning the learning process more difficult.
- The activities that got the best prediction are 0, 1, 3 and 9 (lying, sitting, normal walk and vacuuming); whereas the ones that got the worst prediction are activities 6, 7 and 8 (nordic walk, ascending stairs and descending stairs).

➤ **Subject partially dependent classification:**

The aspects described below were derived from the experiments carried for 11 attributes:

- The best result obtained was an accuracy rate of 70,2267%; versus the worst result, an accuracy rate of 41,7651%. The accuracy mean for all the folds is 52,73%.
- The activities with the best accuracy rate are 0, 10 and 11 (lying, ironing and rope jumping); while the worst accuracy rate was achieved for the activities 5, 6 and 7 (cycling, nordic walk and ascending stairs).
- It is noticeable how frequent the confusion is between the following groups of activities: 1 and 0 (sitting and lying); 2, 9 and 10 (standing, vacuuming and ironing); 3, 7 and 8 (normal walk, ascending stairs and descending stairs); 5 and 6 (cycling and nordic walk); 4 and 11 (running and rope jumping).

In relation to the experiments carried with 22 attributes, these are the observations extracted from the results:

- The best accuracy rate obtained was 67,3025%; whereas the worst accuracy rate was achieved with 39,372%. The mean value for the accuracy rate of all the folds is 51,77%.
- The activities which got the best prediction rate are 0, 1 and 4 (lying, sitting and running); versus activity 11 (rope jumping) that got the worst prediction rate.
- There are strong confusion patterns between the following groups of activities: 0, 1, 2 and 10 (lying, sitting, standing and ironing); 4, 5, 6 and 11 (running, cycling, nordic walk and rope jumping).

Conclusions

Objectives fulfillment

In this chapter, the fulfillment of the objectives defined at the beginning is analyzed:

- ✓ Evolutionary computation techniques have been developed and applied, both an evolutionary algorithm for activities classification and a particle swarm for clustering optimization.
- ✓ Coevolution has been integrated in both parts of the system, getting an efficient use of the available data in the learning phase.
- ✓ A public data set has been used, concretely the PAMAP2 [28], already proposed in other studies, which contains data gathered with wearable sensors.

- ✓ The evaluation of the implemented system has been carried through three different approaches, regarding the subject dependance level of the classifier.
- ✓ The achieved results, while being improvable by expert investigations, also are comparable to the state-of-the-art, in the case of subject dependent classification.
- ✓ A state-of-the-art study has been carried, covering from activities recognition, evolutionary and biological computation, to machine learning.

Regarding a more personal academic aspect, this project has meant:

- The possibility to acquire and apply new knowledge, widening the experience related to what has been learnt during the completion of the degree, in the areas of activities recognition, evolutionary computation and machine learning.
- The opportunity of developing a complete project, which implies the right use of tools and practical application of theoretical knowledge, as well as the convenient documentation of it.

Future work

In this section, some variations to the system developed are suggested, for its improvement in possible future studies:

- Data preprocessing that preserves a higher number of instances. This could be accomplished by substituting the missing attribute values with the ones that are known, instead of removing those instances (1%), as it is done with the heart rate.
- Alternative attribute selection to improve the escalation of the classifiers.
- Application of signal segmentation besides clustering.
- Stratification of training sets when possible.

- Tree pruning method more effective in reducing overfitting, by, for example, getting rid of branches that only lead to wrong predictions.
- The consideration of different attribute weights in clustering phase, based on any magnitude like entropy or information gain.
- Combination of the system modules with other external algorithms.

In conclusion, every goal proposed at the beginning of this project has been accomplished, bringing new lines of investigation in the field of activities recognition, despite the results not being very positive in two out of the three studied approaches. It is another sample of how open is this area, as well as how wide are the applications that evolutionary computation may have, even wider if combined to other disciplines, such as machine learning.

ANEXO II. Árboles

En esta sección, se muestra un ejemplo de cada representación utilizada para la codificación de los árboles-solución: el fenotipo y la codificación simplificada. Concretamente, se recoge la representación de cada tipo de un árbol con un fitness del 99.1438, obtenido sobre su conjunto de entrenamiento.

FENOTIPO

```
atributo6 < 0.6883239273185245
|
|   atributo0 < 0.15580728026670912
|   |
|   |   clase: 1 (16066 correctas)(51 incorrectas)
|   |   atributo6 < 0.5192801691796022
|   |   |
|   |   |   clase: 1 (1492 correctas)(0 incorrectas)
|   |   |   clase: 2 (19363 correctas)(35 incorrectas)
|   atributo4 < 0.6377765210747529
|   |
|   |   clase: 10 (21156 correctas)(0 incorrectas)
|   |   clase: 1 (0 correctas)(273 incorrectas)
```

CODIFICACIÓN SIMPLIFICADA

```
6 0.6883239273185245
0 0.15580728026670912
C 1
6 0.5192801691796022
C 1
C 2
4 0.6377765210747529
C 10
C 1
```

Referencias

- [1] O. D. Lara y M. A. Labrador, "A Survey on Human Activity Recognition using Wearable Sensors", IEEE Communications Surveys & Tutorials, vol. 15, no. 3, Third Quarter 2013.
- [2] A. J. Perez, M. A. Labrador, y S. J. Barbeau, "G-sense: A scalable architecture for global sensing and monitoring", IEEE Network, vol. 24, no. 4, pp. 57–64, 2010.
- [3] O. X. Schmilch, B. Witzschel, M. Cantor, E. Kahl, R. Mehmke, y C. Runge, "Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring", Computers in Human Behavior, vol. 15, no. 5, pp. 571–583, 1999.
- [4] F. Foerster, M. Smeja, y J. Fahrenberg, "Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring", Computers in Human Behavior, vol. 15, no. 5, pp. 571–583, 1999.
- [5] G. Wolf, "Know Thyself: Tracking Every Facet of Life, from Sleep to Mood to Pain, 24/7/365", Wired Magazine, Issue 17.07 – July 2009.
- [6] H. James Wilson. "You, By the Numbers", Harvard Business Review, September 2012 Issue.
- [7] "Counting every moment", The Economist, Mar 3rd 2012.
- [8] T. Hay, "The Rise of the 'Quantified Self' in Health Care", Venture Capital Dispatch, The Wall Street Journal, Aug 13th 2013.
- [9] K. White, "The Quantified Self and Corporate Wellness", Insights & Blogs of Edelman, Feb 22nd 2013.
- [10] J. M. Bradley, "When IoE Gets Personal: The Quantified Self Movement!", Cisco Blog, Sep 10th 2013.
- [11] <http://www.who.int/mediacentre/factsheets/fs385/es/>

[12] <http://www.redbull.com/us/en/bike/stories/1331597139658/project-endurance>

[13] M. Marín, “Los sensores saltan de los ‘wearables’ a la ropa técnica y de uso médico”, TICbeat, 21 marzo 2015.

[14] C. Strohrmann, M. Rossi, B. Arnrich y G. Tröster, “A data-driven approach to kinematic analysis in running using wearable technology”, *Proceedings of the 9th International Conference on Wearable and Implantable Body Sensor Networks (BSN 2012)*, 2012.

[15] C. Strohrmann, J. Seiter, Y. Llorca y G. Tröster, “Can Smartphones Help with Running Technique?”, *The International Workshop on Body Area Sensor Networks (BASNet)*, 2013.

[16] M. Bächlin, K. Förster y G. Tröster, “SwimMaster: A wearable assistant for swimmer”, *UbiComp '09: Proceedings of the 11th international conference on Ubiquitous computing*, Orlando, Florida, USA, pages 215-224, ACM, 2009.

[17] M. Bächlin, M. Kusserow, H. Gubelmann and G. Tröster, “Ski jump analysis of an Olympic champion with wearable acceleration sensors”, *Accepted for Proc. of the 14th International Symposium on Wearable Computers (ISWC 2010)*, 2010.

[18] <http://www.ife.ee.ethz.ch/research/index>

[19] J. Parkka, M. Ermes, P. Korpiä, J. Mantyjarvi, J. Peltola e I. Korhonen, “Activity classification using realistic data from wearable sensors”, *IEEE Trans. Inf. Technol. Biomed.*, 2006.

[20] A. M. Sabatini, “Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing”, *IEEE Trans. Biomed. Eng.*, 2006.

[21] D. Murakami y M. Makikawa, “Ambulatory behavior map, physical activity and biosignal monitoring system”, *Methods Inf. Med.*, 1997.

[22] A. Krause, D. P. Siewiorek, A. Smailagic y J. Farringdon, “Unsupervised, dynamic identification of physiological and activity context in

wearable computing”, Proc. 7th International Symposium on Wearable Computers (White Plains, NY)pp 88–97, 2003.

[23] J. B. Bussmann, J. H. Tulen, E. C. van Herel y H. J. Stam, “Quantification of physical activities by means of ambulatory accelerometry: a validation study”, *Psychophysiology*, 1998.

[24] J. Lester, T. Choudhury y G. Borriello, “A practical approach to recognizing physical activities”, *Pervasive Computing (Lecture Notes in Computer Sciencevol 3968)* (Berlin: Springer) pp 1–16, 2006.

[25] K. van Laerhoven y H. W. Gellersen, “Spine versus porcupine: a study in distributed wearable activity recognition”, Proc. 8th Int. Symposium on Wearable Computers (Arlington) pp 142–50, 2004.

[26] K. van Laerhoven, H. W. Gellersen y Y. G. Malliaris, “Long-term activity monitoring with a wearable sensor node”, *International Workshop on Wearable and Implantable Body Sensor Networks* pp 171–4, 2006.

[27] R. Wijesiriwardana, T. Dias y S. Mukhopadhyay, “Resistive fibre-meshed transducers”, Proc. 7th International Symposium on Wearable Computers (White Plains, NY) pp 200–9, 2003.

[28] A. Reiss y D. Stricker, “Introducing a new benchmarked dataset for activity monitoring”, Proc. Intl. Symp. Wearable Computers (ISWC’12) pp 108-109, 2012.

[29] <http://www.trivisio.com/trivisio-products/colibri-wireless-inertial-motion-tracker-3/>

[30] F. Marin, L. Fradet, K. Lepetit, C. Hansen y K. B. Mansour, “Inertial measurement unit in biomechanics and sport biomechanics: past, present, future”, 33rd International Conference on Biomechanics in Sports, 2015.

[31] I. Maurtua, P. T. Kirisci, T. Stiefmeier, M. L. Sbodio, y H. Witt, “A wearable computing prototype for supporting training activities in automative production”, *Proceedings of the 4th International Forum on Applied Wearable Computing*, 1–12, 2007.

- [32] T. Stiefmeier, D. Roggen, G. Ogris, P. Lukowicz, y G. Troster, "Wearable activity tracking in car manufacturing", *IEEE Pervasive Computing* 7, 2 (2008), 42–50, 2008.
- [33] K. Kunze, M. Barry, E.A. Heinz, P. Lukowicz, D. Majoe, y J. Gutknecht, "Towards recognizing tai chi—An initial experiment using wearable sensors", *Proceedings of FAWC (2006)*, 1–6, 2006.
- [34] D. Minnen, T. Starner, I. Essa, y C. Isbell, "Discovering characteristic actions from on-body sensor data", *Proceedings of the 10th IEEE International Symposium on Wearable Computers (ISWC)*, 2006a.
- [35] C. Ladha, N. Hammerla, P. Olivier y T. Plotz, "ClimbAX: Skill assessment for climbing enthusiasts", *Proceedings of the Int. Conf. Ubiquitous Comp. (UbiComp)*, 2013.
- [36] M. E. Pollack, L. Brown, D. Colbry, C. E. McCarthy, C. Orosz, B. Peintner, S. Ramakrishnan e I. Tsamardinos, "Autominder: an intelligent cognitive orthotic system for people with memory impairment", *Robotics and Autonomous Systems* 44(3-4):273–282, 2003.
- [37] U. Blanke y B. Schiele, "Remember and transfer what you have learned—recognizing composite activities based on activity spotting", *Proceedings of ISWC*, 1–8, 2010.
- [38] M. J. Mathie, A. C. Coster, N. H. Lovell y B. G. Celler, "Detection of daily physical activities using a triaxial accelerometer", *Med. Biol. Eng. Comput.*, 2003.
- [39] A. Bulling, J. A. Ward y H. Gellersse, "Multimodal recognition of reading activity in transit using body-worn sensors", *ACM Transactions on Applied Perception* 9, 1 (2012), 2:1–2:21, 2012.
- [40] K. Kunze y P. Lukowicz, "Dealing with sensor displacement in motion-based onbody activity recognition systems", *Proceedings of UbiComp*, 20–29, 2008.
- [41] H. Bayati, J. d. R. Millan y R. Chavarriaga, "Unsupervised adaptation to on-body sensor displacement ´ in acceleration-based activity recognition", *Proceedings of ISWC*, 2011.

- [42] Z. Yan, V. Subbaraju, D. Chakraborty, A. Misra y Karl Aberer, "Energy-efficient continuous activity recognition on mobile phones: An activity-adaptive approach", Proceedings of ISWC. IEEE, 17–24, 2012.
- [43] T. Gu, Z. Wu, X. Tao, H. K. Pung y J. Lu, "epSICAR: An Emerging Patterns based approach to sequential, interleaved and Concurrent Activity Recognition", Proceedings of the IEEE International Conference on Pervasive Computing and Communications, 2009.
- [44] T. van Kasteren, G. Englebienne, y B. Krse, "An activity monitoring system for elderly care using generative and discriminative models," Journal on Personal and Ubiquitous Computing, 2010.
- [45] A. Tolstikov, X. Hong, J. Biswas, C. Nugent, L. Chen, y G. Parente, "Comparison of fusion methods based on dst and dbn in human activity recognition," Journal of Control Theory and Applications, vol. 9, pp. 18– 27, 2011.
- [46] C. N. Joseph, S. Kokulakumaran, K. Srijevanth, A. Thusyanth, C. Gunasekara y C. Gamage, "A framework for whole-body gesture recognition from video feeds," in International Conference on Industrial and Information Systems (ICIIS), pp. 430–435, 2010.
- [47] P. Turaga, R. Chellappa, V. Subrahmanian y O. Udrea, "Machine recognition of human activities: A survey," IEEE Transactions on Circuits and Systems for Video Technology, vol. 18, no. 11, pp. 1473– 1488, 2008.
- [48] J. Candamo, M. Shreve, D. Goldgof, D. Sapper y R. Kasturi, "Understanding transit scenes: A survey on human behavior-recognition algorithms," IEEE Transactions on Intelligent Transportation Systems, vol. 11, no. 1, pp. 206–224, 2010.
- [49] L. Piyathilaka, S. Kodagoda, "Gaussian mixture based HMM for human daily activity recognition using 3D skeleton features", 8th IEEE Conference on Industrial Electronics and Applications (ICIEA), 2013
- [50] T. Choudhury, G. Borriello et al., "The Mobile Sensing Platform: An Embedded System for Activity Recognition", IEEE Pervasive Magazine - Special Issue on Activity-Based Computing, April 2008.

- [51] N. Ravi, N. Dandekar, P. Mysore y M. Littman, "Activity Recognition from Accelerometer Data", Proceedings of the Seventeenth Conference on Innovative Applications of Artificial Intelligence (IAAI/AAAI 2005).
- [52] R. Want, A. Hopper, V. Falcao y J. Gibbons, "The Active Badge Location System", ACM Transactions on Information Systems, Vol. 40, No. 1, pp. 91-102, January 1992.
- [53] T. Gu, Z. Wu, L. Wang, X. Tao y J. Lu, "Mining Emerging Patterns for Recognizing Activities of Multiple Users in Pervasive Computing", Proceedings of the 6th International Conference on Mobile and Ubiquitous Systems: Computing Networking and Services (MobiQuitous '09), Toronto, Canada, July 13–16, 2009.
- [54] D. Gordon, J. Hanne, M. Berchtold, A. Asghar, N. Shirehjini y M. Beigl, "Towards Collaborative Group Activity Recognition Using Mobile Devices", Mobile Networks and Applications 18(3), p. 326-340, 2013.
- [55] T. Hirano y T. Maekawa, "A hybrid unsupervised/supervised model for group activity recognition", Proceedings of the 2013 International Symposium on Wearable Computers, ISWC '13, ACM (New York, NY, USA, 2013), 21–24, 2013.
- [56] O. Brdiczka, J. Maisonnasse, P. Reignier y J. L. Crowley, "Detecting small group activities from multimodal observations", Applied Intelligence 30, 1 (July 2007), 47–57.
- [57] E. M. van Sluijs, A. M. McMinn y S. J. Griffin, "Effectiveness of interventions to promote physical activity in children and adolescents: systematic review of controlled trials", Br. Med. J, 2007.
- [58] K. Aminian, P. Robert, E. E. Buchser, B. Rutschmann, D. Hayoz y M. Depairon, "Physical activity monitoring based on accelerometry: validation and comparison with video observation", Med. Biol. Eng. Comput, 1999.
- [59] D. Maxwell, "Addressing the challenge of quantifying free-living activity—the activPAL™ professional", Conf. on Recent Advances in Assistive Technology and Engineering (RAATE) (Birmingham)p23, 2002.
- [60] B. Najafi, K. Aminian, F. Loew, Y. Blanc y P. A. Robert, "Measurement of stand-sit and sit-stand transitions using a miniature gyroscope and its

application in fall risk evaluation in the elderly", IEEE Trans. Biomed. Eng, 2002.

[61] B. Najafi, K. Aminian, A. Paraschiv-Ionescu, F. Loew, C. J. Bula y P. Robert, "Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly", IEEE Trans. Biomed. Eng, 2003.

[62] J. Baek, G. Lee, W. Park y B. J. Yun, "Accelerometer signal processing for user activity detection", Knowledge-Based Intelligent Information and Engineering Systems, Pt 3, Proceedings, pp 610–7, 2004.

[63] R. Herren, A. Sparti, K. Aminian e Y. Schutz, "The prediction of speed and incline in outdoor running in humans using accelerometry", Med. Sci. Sports Exerc., 1999.

[64] U. Maurer, A. Rowe, A. Smailagic y D. Siewiorek, "Location and activity recognition using eWatch: a wearable sensor platform", Ambient Intelligence in Everyday Life (Lecture Notes in Computer Science vol 3864) (Berlin: Springer) pp 86–102, 2006.

[65] K. Aminian, P. Robert, E. Jequier e Y. Schutz, "Incline, speed, and distance assessment during unconstrained walking", Med. Sci. Sports Exerc. 27:226–34, 1995.

[66] F. Foerster y J. Fahrenberg, "Motion pattern and posture: correctly assessed by calibrated accelerometers", Behav. Res. Methods Instrum. Comput., 2000.

[67] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney y D. Howard, "A comparison of different feature generation methods in activity classification", Proc. ICAMPAM (Rotterdam) p87, 2008.

[68] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney y D. Howard, "A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data", IEEE Trans. Biomed. Eng. at press, 2008.

[69] T. Huynh y B. Schiele, "Analyzing features for activity recognition", Proc. Conf. Smart Objects and Ambient Intelligence: Innovative Context-Aware Services: Usages and Technologies (Grenoble) pp 159–64, 2005.

[70] A. Sugimoto, Y. Hara, T. W. Findley y K. Yoncmoto, "A useful method for measuring daily physical activity by a three-direction monitor", *Scand. J. Rehabil. Med.* 29:37–42, 1997.

[71] L. Bao y S. S. Intille, "Activity recognition from user-annotated acceleration data", *Pervasive Computing (Lecture Notes in Computer Science vol 3001)* (Berlin: Springer) pp 1–17, 2004.

[72] A. Paraschiv-Ionescu, E. E. Buchser, B. Rutschmann, B. Najafi y K. Aminian, "Ambulatory system for the quantitative and qualitative analysis of gait and posture in chronic pain patients treated with spinal cord stimulation", *Gait Posture*, 2004.

[73] M. N. Nyan, F. E. Tay, K. H. Seah e Y. Y. Sitoh, "Classification of gait patterns in the time-frequency domain", *J. Biomech*, 2006.

[74] M. Sekine, T. Tamura, T. Togawa e Y. Fukui, "Classification of waist-acceleration signals in a continuous walking record", *Med. Eng. Phys.*, 2000.

[75] M. Sekine, T. Toshiyo, F. Toshiro y F. Yasuhiro, "Classification of walking pattern using acceleration waveform in elderly people", 22nd Annual Conf. of the IEEE Engineering in Medicine and Biology Society (Chicago) pp 1356–9, 2000.

[76] M. Sekine, T. Tamura, M. Akay, T. Fujimoto, T. Togawa e Y. Fukui, "Discrimination of walking patterns using wavelet-based fractal analysis *IEEE Trans. Neural Syst. Rehabil. Eng.*, 2002.

[77] T. Tamura, M. Sekine, M. Ogawa, T. Togawa e Y. Fukui, "Classification of acceleration waveforms during walking by wavelet transform", *Methods Inf. Med.* 36:356–9, 1997.

[78] N. Wang, E. Ambikairajah, N. H. Lovell y B. G. Celler, "Accelerometry based classification of walking patterns using time-frequency analysis", 29th Annual Conf. of the IEEE Engineering in Medicine and Biology Society (Lyon) pp 4899–902, 2007.

[79] K. Kiani, C. J. Snijders y E. S. Gelsema, "Computerized analysis of daily life motor activity for ambulatory monitoring", *Technol. Health Care* 5:307–18, 1997.

- [80] P. Pirttikangas, K. Fujinami y T. Nakajima, "Feature selection and activity recognition from wearable sensors", *Ubiquitous Computing Systems (Lecture Notes in Computer Science vol 4239)* (Berlin: Springer) pp 516–27, 2006.
- [81] T. Plotz, N. Y. Hammerla y P. Olivier, "Feature learning for activity recognition in ubiquitous computing", *Proceedings of the 22nd international Joint Conference on Artificial Intelligence*, AAAI Press, 1729–1734, 2011.
- [82] T. Chau, "A review of analytical techniques for gait data: part 1. Fuzzy, statistical and fractal methods", *Gait Posture*, 2001.
- [83] R. O. Duda, P. E. Hart y D. G. Stork, "Pattern Classification", 2nd edn (New York: Wiley), 2001.
- [84] A. Webb, "Statistical Pattern Recognition" (Chichester: Wiley), 2002.
- [85] T. Huynh y B. Schiele, "Unsupervised discovery of structure in activity data using multiple eigenspaces", *Second International Workshop on Location- and Context-Awareness (Dublin)* pp 151–67, 2006.
- [86] J. Yin, Q. Yang, and J. Pan, "Sensor-based abnormal human-activity detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 8, pp. 1082–1090, 2008.
- [87] O. D. Lara, A. J. Perez, M. A. Labrador, and J. D. Posada, "Centinela: A human activity recognition system based on acceleration and vital sign data," *Journal on Pervasive and Mobile Computing*, 2011.
- [88] M. Berchtold, M. Budde, H. Schmidtke, and M. Beigl, "An extensible modular recognition concept that makes activity recognition practical," in *Advances in Artificial Intelligence, Lecture Notes in Computer Science*, pp. 400–409, Springer Berlin / Heidelberg, 2010.
- [89] J. Boyle, T. Karunanithi, T. Wark, W. Chan y C. Colavitti, "Quantifying functional mobility progress for chronic disease management", *28th Annual Conf. of the IEEE Engineering in Medicine and Biology Society (New York)* pp 5916–9, 2006.

- [90] K. M. Culhane, G. M. Lyons, D. Hilton, P. A. Grace y D. Lyons, "Long-term mobility monitoring of older adults using accelerometers in a clinical environment", *Clin. Rehabi.*, 2004
- [91] M. Uiterwaal, E. B. Glerum, H. J. Busser y R. C. van Lummel, "Ambulatory monitoring of physical activity in working situations, a validation study", *J. Med. Eng. Technol.* 22168–72, 1998.
- [92] H. J. Busser, J. Ott, R. C. van Lummel, M. Uiterwaal y R. Blank, "Ambulatory monitoring of children's activity", *Med. Eng. Phys.*, 1997.
- [93] J. B. J. Bussmann, Y. M. van de Laar, M. P. Neeleman y H. J. Stam, "Ambulatory accelerometry to quantify motor behaviour in patients after failed back surgery: a validation study", *Pain*, 1998.
- [94] M. Makikawa y H. Iizumi, "Development of an ambulatory physical activity memory device and its application for the categorization of actions in daily life", *Medinfo* 8747–50, 1995.
- [95] P. H. Veltink, H. B. Bussmann, W. de Vries, W. L. Martens y R. C. Van Lummel, "Detection of static and dynamic activities using uniaxial accelerometers", *IEEE Trans. Rehabil. Eng.*, 1996.
- [96] B. Coley, B. Najafi, A. Paraschiv-Ionescu y K. Aminian, "Stair climbing detection during daily physical activity using a miniature gyroscope", *Gait Posture*, 2005.
- [97] K. Aminian, B. Najafi, C. Bula, P. F. Leyvraz y P. Robert, "Spatio-temporal parameters of gait measured by an ambulatory system using miniature gyroscopes", *J. Biomech.*, 2002.
- [98] M. J. Mathie, B. G. Celler, N. H. Lovell y A. C. Coster, "Classification of basic daily movements using a triaxial accelerometer", *Med. Biol. Eng. Comput.*, 2004.
- [99] J. Fahrenberg, F. Foerster, M. Smeja y W. Muller, "Assessment of posture and motion by multichannel piezoresistive accelerometer recordings", *Psychophysiology* 34607–12, 1997.
- [100] J. Fahrenberg, W. Muller, F. Foerster y M. Smeja, "A multi-channel investigation of physical activity", *Psychophysiology* 10209–17, 1996.

- [101] S. H. Lee, H. D. Park, S. Y. Hong, K. J. Lee e Y. H. Kim, "A study on the activity classification using a triaxial accelerometer", 25th Annual Conf. of the IEEE Engineering in Medicine and Biology Society (Cancun) pp 2941–3, 2003.
- [102] M. Ermes, J. Parkka, J. Mantyjarvi e I. Korhonen, "Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions", IEEE Trans. Inf. Tech. Biomed, 2008.
- [103] J. B. Bussmann, W. L. Martens, J. H. Tulen, F. C. Schasfoort, H. J. Van Den Berg-Emons y H. J. Stam, "Measuring daily behavior using ambulatory accelerometry: the activity monitor", Behav. Res. Methods Instrum. Comput. 33 349–56, 2001.
- [104] N. Cristianini y J. Shawe-Taylor, "An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods", (Cambridge: Cambridge University Press), 2000.
- [105] V. N. Vapnik, "Statistical Learning Theory", (NJ: Wiley), 1998.
- [106] T. Huynh y B. Schiele, "Towards less supervision in activity recognition from wearable sensors", Proc. of the 10th IEEE International Symposium on Wearable Computers (Montreaux) pp 3–10, 2006.
- [107] A. Krause, M. Ihmig, E. Rankin, D. Leong, S. Gupta, D. Siewiorek, A. Smailagic, M. Deisher y U. Sengupta, "Trading off prediction accuracy and power consumption for context-aware wearable computing", Proc. of the 9th IEEE International Symposium on Wearable Computers, pp 20–6, 2005.
- [108] J. Y. Goulermas, A. H. Findlow, C. J. Nester, P. Liatsis, X. J. Zeng, L. P. J. Kenney, P. Tresadern, S. B. Thies y D. Howard, "An instance-based algorithm with auxiliary similarly information for the estimation of gait kinematics from wearable sensors", IEEE Trans. Neural Networks 19 1574–82, 2008.
- [109] J. Y. Goulermas, D. Howard, C. J. Nester, R. K. Jones y L. Ren, "Regression techniques for the prediction of lower limb kinematics", J. Biomech. Eng, 2005.
- [110] C. M. Bishop, "Neural Networks for Pattern Recognition", (Oxford: Oxford University Press), 1999.

- [111] S. Haykin, "Neural Networks; a Comprehensive Foundation, (Englewood Cliffs, NJ: Prentice-Hall), 1999.
- [112] K. Zhang, M. Sun, D. K. Lester, F. X. Pi-Sunyer, C. N. Boozer y R. W. Longman, "Assessment of human locomotion by using an insole measurement system and artificial neural networks", J. Biomech, 2005.
- [113] J. Mantyjarvi, J. Himberg y T. Seppanen, "Recognizing human motion with multiple acceleration sensors", IEEE International Conference on Systems, Man, and Cybernetics (Tucson), pp 747–52, 2001.
- [114] N. Kern, B. Schiele y A. Schmidt, "Multi-sensor activity context detection for wearable computing", European Symposium on Ambient Intelligence (Eindhoven) pp 220–32, 2003.
- [115] W. H. Wu, A. A. Bui, M. A. Batalin, D. Liu y W. J. Kaiser, "Incremental diagnosis method for intelligent wearable sensor systems", IEEE Trans. Inf. Technol. Biomed, 2007.
- [116] F. R. Allen, E. Ambikairajah, N. H. Lovell y B. G. Celler, "Classification of a known sequence of motions and postures from accelerometry data using adapted Gaussian mixture models", Physiol. Meas., 2006.
- [117] D. M. Pober, J. Staudenmayer, C. Raphael y P.S. Freedson, "Development of novel techniques to classify physical activity mode using accelerometers", Med. Sci. Sports Exerc, 2006.
- [118] J. Lester, T. Choudhury, N. Kern, G. Borriello y B. Hannaford, "A hybrid discriminative/generative approach for modeling human activities", 19th International Joint Conference on Artificial Intelligence (Edinburgh) pp 766–72, 2005.
- [119] J. A. Ward, P. Lukowicz, G. Troster y T. E. Starner, "Activity recognition of assembly tasks using body-worn microphones and accelerometers", IEEE Trans. Pattern Anal., 2006.
- [120] S. Lee y K. Mase, "Activity and location recognition using wearable sensors", IEEE Perv. Comp. 125–32, 2002.

- [121] A. Salarian, H. Russmann, F. J. G. Vingerhoets, P. R. Burkhard y K. Aminian, "Ambulatory monitoring of physical activities in patients with Parkinson's disease", IEEE Trans. Biomed. Eng., 2007.
- [122] M. Stikic, D. Larlus, y B. Schiele, "Multi-graph based semi-supervised learning for activity recognition," in International Symposium on Wearable Computers, pp. 85–92, 2009.
- [123] M. Stikic, D. Larlus, S. Ebert, y B. Schiele, "Weakly supervised recognition of daily life activities with wearable sensors," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 12, pp. 2521–2537, 2011.
- [124] A. Ali, R. King, y G.-Z. Yang, "Semi-supervised segmentation for activity recognition with multiple eigenspaces," in International Summer School and Symposium on Medical Devices and Biosensors, pp. 314– 317, 2008.
- [125] D. Guan, W. Yuan, Y.-K. Lee, A. Gavrilov, and S. Lee, "Activity recognition based on semi-supervised learning," in 13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, pp. 469–475, 2007.
- [126] K. van Laerhoven y O. Cakmakci, "What shall we teach our pants?", Proc 4th Int. Symposium on Wearable Computers, pp 77–83, 2000.
- [127] "Datasets for Human Activity Recognition from Tim Van Kasteren's website," <https://sites.google.com/site/tim0306/datasets>.
- [128] "Datasets for Human Activity Recognition from MIT Media Lab," <http://architecture.mit.edu/house n/data/PlaceLab/PlaceLab.htm>.
- [129] "Datasets for Human Activity Recognition from Tanzeem Choudhury's website," <http://www.cs.dartmouth.edu/~tanzeem/teaching/CS188-Fall08/dataset.html>.
- [130] "Datasets for Human Activity Recognition from ETH's Wearable Computing Lab," <http://www.wearable.ethz.ch/resources/Dataset>.

- [131] “PAMAP2 Physical Activity Monitoring Data Set”, <https://archive.ics.uci.edu/ml/datasets/PAMAP2+Physical+Activity+Monitoring>
- [132] “2011 Activity Recognition Challenge,” www.oportunity-project.eu/challenge
- [133] R. Helaoui, M. Niepert, and H. Stuckenschmidt, “Recognizing interleaved and concurrent activities: A statistical-relational approach,” in IEEE International Conference on Pervasive Computing and Communications (PerCom), pp. 1–9, 2011.
- [134] I. H. Witten, E. Frank y M. A. Hall, “Data Mining: Practical Machine Learning Tools and Techniques”, Morgan Kaufmann, 3 ed., 2011.
- [135] C. Elkan, “The foundations of cost-sensitive learning,” in Seventeenth International Joint Conference on Artificial Intelligence, pp. 973–978, 2001.
- [136] L. Vinh, S. Lee, H. Le, H. Ngo, H. Kim, M. Han, y Y.-K. Lee, “Semi-markov conditional random fields for accelerometer-based activity recognition,” Applied Intelligence, vol. 35, pp. 226–241, 2011.
- [137] L. Dugdill, D. Crone y R. Graham, “Physical Activity and Health Promotion: Evidence-Based Approaches to Practice”, (Oxford: Wiley-Blackwell), 2009.
- [138] R. J. Dunnewold, C. E. Jacobi y J. J. van Hilten, “Quantitative assessment of bradykinesia in patients with Parkinson’s disease”, J. Neurosci. Methods, 1997.
- [139] A. V. Ng y J. A. Kent-Braun, “Quantitation of lower physical activity in persons with multiple sclerosis”, Med. Sci. Sports Exerc. 29:517–23, 1997.
- [140] F. Pitta, T. Troosters, V. S. Probst, M. A. Spruit, M. Decramer y R. Gosselink, “Quantifying physical activity in daily life with questionnaires and motion sensors”, COPD Eur. Respir. J. 27:1040–55, 2006.
- [141] N. C. Barengo, G. Hu, T. A. Lakka, H. Pekkarinen, A. Nissinen y J. Tuomilehto, “Low physical activity as a predictor for total and cardiovascular disease mortality in middle-aged men and women”, Finland Eur. Heart J., 2004.

- [142] S. N. Blair, T. R. Collingwood, R. Reynolds, M. Smith, R. D. Hagan y C. L. Sterling, "Health promotion for educators: impact on health behaviors, satisfaction, and general well-being", *Am. J. Public Health*, 1984.
- [143] J. E. Manson, E. B. Rimm, M. J. Stampfer, G. A. Colditz, W. C. Willett, A. S. Krolewski, B. Rosner, C. H. Hennekens y F. E. Speizer, "Physical activity and incidence of non-insulin-dependent diabetes mellitus in women", *Lancet*, 1991.
- [144] A. K. Yancey, C. M. Wold, W. J. McCarthy, M. D. Weber, B. Lee, P. A. Simon y J. E. Fielding, "Physical inactivity and overweight among Los Angeles County adults", *Am.J.Prev.Med*, 2004.
- [145] A. Baldominos, Y. Saez y P. Isasi, "Feature Set Optimization for Physical Activity Recognition Using Genetic Algorithms", *Proceedings of the Companion Publication of the 2015 on Genetic and Evolutionary Computation Conference GECCO COMpanion '15*, Pages 1311-1318, 2015.
- [146] Y. Saeys, I. Inza y P. Larrañaga, "A review of feature selection techniques in bioinformatics", *Bioinformatics*, 23(19):2507-2517, 2007.
- [147] A. Reiss y D. Stricker, "Personalized mobile physical activity recognition", *Proceedings of 17th International Symposium on Wearable Computers (ISWC)*, pp. 25-28, Zurich, Switzerland, September 2013.
- [148] A. Reiss, G. Hendeby y D. Stricker, "Towards robust activity recognition for everyday life: Methods and evaluation", *Proceedings of 7th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, pp. 25-32, Venice, Italy, May 2013.
- [149] A. Reiss y D. Stricker, "Aerobic activity monitoring: Towards a long-term approach", *International Journal of Universal Access in the Information Society (UAIS)*, pp. 1-14, March 2013.
- [150] G. Bleser, D. Steffen, A. Reiss, M. Weber, G. Hendeby y L. Fradet, "Personalized Physical Activity Monitoring Using Wearable Sensors", *Smart Health* 2015: 99-124, 2015.
- [151] <http://www.pamap.org/index.html>

- [152] A. Reiss y D. Stricker, "Creating and Benchmarking a New Dataset for Physical Activity Monitoring", The 5th Workshop on Affect and Behaviour Related Assistance (ABRA), 2012.
- [153] E. Hoque y J. Stankovic, "AALO: Activity recognition in smart homes using Active Learning in the presence of Overlapped activities", 2012 6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), 21-24 May 2012.
- [154] The 19th International Symposium on Wearable Computers, <http://www.iswc.net/iswc15/>
- [155] L. T. Nguyen, M. Zeng, P. Tague y J. Zhang, "Recognizing New Activities with Limited Training Data", ISWC '15 Proceedings of the 2015 ACM International Symposium on Wearable Computers, Pages 67-74, 2015.
- [156] J. R. Kwapisz, G. M. Weiss y S. A. Moore, "Activity Recognition using Cell Phone Accelerometers", ACM SIGKDD Explorations Newsletter archive, Volume 12 Issue 2, December 2010.
- [157] E. M. Tapia, S. S. Intille, W. Haskell, K. Larson, J. Wright, A. King y R. Friedman, "Real-Time Recognition of Physical Activities and Their Intensities Using Wireless Accelerometers and a Heart Rate Monitor", 2007 11th IEEE International Symposium on Wearable Computers, 11-13 Oct. 2007.
- [158] E. Kim, S. Helal y D. Cook, "Human activity recognition and pattern discovery", Pervasive Computing, IEEE, 2010.
- [159] Andreas Bulling, Ulf Blanke, and Bernt Schiele. 2014. A tutorial on human activity recognition using bodyworn inertial sensors. ACM Comput. Surv. 46, 3, Article 33 (January 2014), 33 pages.
- [160] S. J. Preece, J. Y. Goulermas, L. P. J. Kenney, D. Howard, K. Meijer y R. Crompton, "Activity identification using body-mounted sensors—a review of classification techniques", 2009 Institute of Physics and Engineering in Medicine, Physiological Measurement, Volume 30, Number 4, 2 April 2009.
- [161] A. Reiss, G. Hendebay, G. Bleser y D. Stricker, "Activity recognition using biomechanical model based pose estimation", Proceedings of 5th

European Conference on Smart Sensing and Context (EuroSSC), pp. 42-55, Passau, Germany, November 2010.

[162] J. R. Quinlan, "Induction of decision trees", Machine Learning, 1819, 81-106, 1986.

[163] J. R. Quinlan, "Learning with continuous classes", N. Adams & L. Sterling (Eds.), Proceedings of the Fifth Australian Joint Conference on Artificial Intelligences (pp. 343-348), Hobart, Tasmania; Singapore: World Scientific; 1992.

[164] J. R. Quinlan, "C4.5: Programs for machine learning", San Francisco: Morgan Kaufmann, 1993.

[165] J. R. Quinlan, "Improved use of continuous attributes in C4.5", Journal of Artificial Intelligence Research, 4, 77-90, 1996.

[166] C. E. Shannon, "A Mathematical Theory of Communication", Bell System Technical Journal, 27, pp. 379-423 & 623-656, July & October, 1948.

[167] A. R. Wallace, "On the Tendency of Varieties to Depart Indefinitely From the Original Type", Febrero 1858, tercera parte de "On the Tendency of Species to form Varieties; and on the Perpetuation of Varieties and Species by Natural Means of Selection." [170], pp 53-62.

[168] C. Darwin, "Extract from an unpublished Work on Species", porción del capítulo "On the Variation of Organic Beings in a state of Nature; on the Natural Means of Selection; on the Comparison of Domestic Races and true Species.", del ensayo escrito en 1844. Primera parte de "On the Tendency of Species to form Varieties; and on the Perpetuation of Varieties and Species by Natural Means of Selection." [170], pp 46-50.

[169] C. Darwin, "Abstract of a Letter to Prof. ASA GRAY", Boston, U.S., September 5th, 1857. Segunda parte de "On the Tendency of Species to form Varieties; and on the Perpetuation of Varieties and Species by Natural Means of Selection." [170], pp 50-53.

[170] C. Darwin y A. R. Wallace, "On the Tendency of Species to form Varieties; and on the Perpetuation of Varieties and Species by Natural Means of Selection.", comunicado por C. Lyell y J. D. Hooker a la LSL el 1

de julio de 1858; publicado en Journal of the Proceedings of the Linnean Society of London: Zoology 3, Issue 9, pages 45–62, 20 Agosto 1858.

[171] C. Darwin, “On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life.”, London: John Murray. [1st ed.], November 24, 1859.

[172] G. Mendel, “Versuche über Pflanzenhybriden.”, Verhandlungen des naturforschenden Vereines in Brünn, Bd. IV für das Jahr 1865, Abhandlungen, 3–47, 1866. (“*Experimentos sobre hibridación de plantas*”, *Sociedad de Historia Natural de Brno*), leído en las reuniones del 8 de febrero y el 8 de marzo de 1865.

[173] L. J. Fogel, A. J. Owens y M. J. Walsh, “Artificial Intelligence through Simulated Evolution”, New York: John Wiley & Sons, 1966.

[174] L. J. Fogel, A. J. Owens y M. J. Walsh, “Intelligent Decision Making Through a Simulation of Evolution”, *Behavioral Science* **11** (4): 253–272, 1966.

[175] L. J. Fogel, “Intelligence through Simulated Evolution: Forty Years of Evolutionary Programming”, New York: John Wiley & Sons, 1999.

[176] L. J. Fogel, “Autonomous Automata”. *Industrial Research* **4** (2): 14–19, 1962.

[177] L. J. Fogel, A. J. Owens y M. J. Walsh, “Adaption of Evolutionary Programming to the Prediction of Solar Flares, No. N-66-21096, NASA-CR-417”. NASA. 1966.

[178] L. J. Fogel, A. J. Owens y M. J. Walsh,). “On the Evolution of Artificial Intelligence”. *Proceedings of the Fifth National Symposium on Human Factors in Electronics*, San Diego, CA May 5–6 (IEEE): 63–76, 1964.

[179] L. J. Fogel, A. J. Owens y M. J. Walsh, “An Evolutionary Prediction Technique”. *IEEE International Symposium on Microwaves, Circuit Theory, and Information Theory*, September (IEEE): 173–174, 1964.

[180] I. Rechenberg, “Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipien der biologischen Evolution” PhD thesis, 1971. Reprinted by Fromman-Holzboog (1973).

[181] H. Schwefel, "Numerische Optimierung von Computer-Modellen", PhD thesis, 1974. Reprinted by Birkhäuser (1977).

[182] H. Schwefel, "Evolution and Optimum Seeking", New York: John Wiley & Sons, 1995.

[183] I. Rechenberg "Evolutionsstrategie '94", Stuttgart: Frommann-Holzboog, 1994.

[184] J. H. Holland, "Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence.", University of Michigan Press, 1975.

[185] J. R. Koza, "Genetic programming: on the programming of computers by means of natural selection", Vol. 1, MIT press, 1992.

[186] M. Minsky, "The Society of Mind", Simon and Schuster, 1988.

[187] H. Moravec, "Mind Children: The Future of Robot and Human Intelligence", Harvard University Press, September 1988.

[188] H. Moravec, "Robot: mere machine to transcendent mind", Oxford University Press, November 1998.

[189] R. A. Brooks, "Elephants don't play chess", Robotics and Autonomous Systems, Volume 6 Issue 1-2, Pages 3-15, June 1990.

[190] <http://iridia.ulb.ac.be/comp2sys/research-ec.php.html>

[191] F. J. Vico Vela, "Computación evolutiva: El legado de Darwin en la Ingeniería Informática", UMA. SEDOC, Uciencia : revista de divulgación científica de la Universidad de Málaga, Nº 1 (Julio), pp 39-41, 2009.

[192] <http://tracer.uc3m.es/tws/cEA/index.htm>

[193] R. C. Eberhart y J. Kennedy, "A new optimizer using particle swarm theory", Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan. pp. 39-43, 1995.

[194] J. Kennedy y R. C. Eberhart, "Particle swarm optimization", Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ. pp. 1942-1948, 1995.

[195] J. Kennedy y R. C. Eberhart, "A discrete binary version of the particle swarm algorithm", Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics 1997, Piscataway, NJ. pp. 4104-4109, 1997.

[196] <http://www.swarmintelligence.org/index.php>

[197] <http://tracer.uc3m.es/tws/psa/index.html>

[198] <http://iridia.ulb.ac.be/~mdorigo/ACO/about.html>

[199] M. Dorigo, "Optimization, Learning and Natural Algorithms", Ph.D.Thesis, Politecnico di Milano, Italy, [in Italian], 1992.

[200] M. Dorigo y L.M. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem", Technical Report TR/IRIDIA/1996-5, IRIDIA, Université Libre de Bruxelles, 1996. Later published in IEEE Transactions on Evolutionary Computation, 1(1):53-66, 1997.

[201] L.M. Gambardella y M. Dorigo, "Ant-Q: A Reinforcement Learning Approach to the Traveling Salesman Problem", Proceedings of ML-95, Twelfth International Conference on Machine Learning, Tahoe City, CA, A. Prieditis and S. Russell (Eds.), Morgan Kaufmann, 252-260, 1995.

[202] T. Stützle y H. H. Hoos, "Improving the Ant System: A detailed report on the MAX-MIN Ant System", Technical report AIDA-96-12, FG Intellektik, FB Informatik, TU Darmstadt, 1996. Later published in part as "The Max-Min Ant System and Local Search for the Travelling Salesman Problem", IEEE International Conference on Evolutionary Computation, Piscataway, T. Bäck, Z. Michalewicz and X. Yao (Eds.), IEEE Press, pp. 309-314, 1997.

[203] M. Dorigo y G. Di Caro, "The Ant Colony Optimization Meta-Heuristic", In D. Corne, M. Dorigo and F. Glover, editors, New Ideas in Optimization, McGraw-Hill, 11-32, 1999.

[204] M. Dorigo, G. Di Caro y L. M. Gambardella, "Ant Algorithms for Discrete Optimization", Artificial Life, 5(2):137-172, 1999.

- [205] M. Dorigo, V. Maniezzo y A. Colorni, "Ant System: Optimization by a colony of cooperating agents", IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26(1):29-41, 1996.
- [206] P. Ehrlich y P. Raven, "Butterflies and plants: A study in coevolution", Evolution, 18:586-608, 1965.
- [207] L. Van Valen, "A new evolutionary law", Evolutionary Theory 1, 1-30, 1973.
- [208] R. Dirzo y J. N. Thompson, "La coevolución y las enseñanzas de Darwin", Ciencia hoy, ISSN 0327-1218, Vol. 19, N°. 113, págs. 43-48, (OCT-NOV), 2009.
- [209] C. A. Coello, "Introducción a la Computación Evolutiva", Departamento de Computación CINVESTAV-IPN, <http://delta.cs.cinvestav.mx/~ccoello/compevol/clase14-2015.pdf>.
- [210] M. A. Potter y K. A. De Jong, "A cooperative coevolutionary approach to function optimization," in PPSN III: Proceedings of the International Conference on Evolutionary Computation, pp. 249-257. The Third Conference on Parallel Problem Solving from Nature. London, UK: Springer-Verlag, 1994.
- [211] R. P. Wiegand, W. C. Liles y K. A. De Jong, "An empirical analysis of collaboration methods in cooperative coevolutionary algorithms.", Proceedings of the genetic and evolutionary computation conference (GECCO). Vol. 2611. July 2001.
- [212] D. Wallin, C. Ryan, y R. Azad, "Symbiogenetic coevolution", The 2005 IEEE Congress on Evolutionary Computation, 2005. Vol. 2. IEEE, Sept 2005.
- [213] K. Adamu y S. Phelps, "An empirical study of collaboration methods for coevolving technical trading rules", 2010 UK Workshop on Computational Intelligence (UKCI) 1-6, 2010.
- [214] C. K. Au y H. F. Leung, "On the behavior of cooperative coevolution in dynamic environments", In IEEE Congress on Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence), pp. 2827-2836. IEEE, June 2008.

[215] K. Sims, "Evolving 3D Morphology and Behavior by Competition", Artificial Life IV (in this volume), R. Brooks and Pattie Maes, Eds., MIT Press, Cambridge, Massachusetts, 1994.

[216] J. Paredis, "Coevolutionary computation", Artificial life 2.4: 355-375, 1995.

[217] J. Paredis, "Coevolutionary process control", Artificial Neural Nets and Genetic Algorithms. Springer Vienna, 1998.

[218] C. D. Rosin y R. K. Belew, "New methods for competitive coevolution", Evolutionary Computation 5, no. 1: 1-29, 1997.

[219] C. D. Rosin, "Coevolutionary search among adversaries", PhD diss., University of California, San Diego, 1997.

[220] T. Miconi, y A. Channon, "An improved system for artificial creatures evolution", Procs 10th Intl Conf on Simulation and Synthesis of Living Systems (ALIFE X). 2006.

[221] J. Togelius, P. Burrow y S. M. Lucas, "Multi-population competitive co-evolution of car racing controllers", Proceedings of the IEEE Congress on Evolutionary Computation (CEC), 4043-4050, 2007.

[222] <http://ocw.uc3m.es/ingenieria-informatica/computacion-biologica/material-de-clase>

[223] J. Kennedy, R. C. Eberhart y Y. Shi, "Swarm intelligence", Morgan Kaufmann Publishers, San Francisco, 2001.

[224] J. Kennedy y R. C. Eberhart, "A discrete binary version of the particle swarm algorithm", In Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics, pp 4104-4109, 1997.

[225] X. Hu, Y. Shi y R. C. Eberhart, "Recent advances in particle swarm", In Proceedings of IEEE Congress on Evolutionary Computation 2004 (CEC 2004), pages 90-97, 2004.

[226] S. Yang, M. Wang y L. Jiao, "A quantum particle swarm optimization", In Proceedings of IEEE Congress on Evolutionary Computation 2004 (CEC 2004), pages 320-331, 2004.

[227] T. Sousa, A. Silva y A. Neves, "Particle swarm based data mining algorithms for classification tasks", *Parallel Comput.*, 30(5-6):767-783, 2004.

[228] M. Clerc, Binary particle swarm optimisers: toolbox, derivations and mathematical insights. http://clerc.maurice.free.fr/pso/binary_pso.

[229] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann e I. H. Witten, "The WEKA Data Mining Software: An Update", *SIGKDD Explorations*, Volume 11, Issue 1, 2009.

[230] Weka 3: Data Mining Software in Java: <http://www.cs.waikato.ac.nz/ml/weka/>

[231] Open Source Initiative FAQ: <https://opensource.org/faq>

[232] Free Software Definition: <http://www.gnu.org/philosophy/free-sw.html>

[233] L. Sweeney, "Simple Demographics Often Identify People Uniquely", Carnegie Mellon University, Data Privacy Working Paper 3, Pittsburgh, 2000.